TWO DIMENSIONAL MATCHED FILTERING FOR ATTITUDE MEASUREMENT WITH APPLICATIONS TO TARGET TRACKING

Robert Howard Williams

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

TWO DIMENSIONAL MATCHED FILTERING
FOR ATTITUDE MEASUREMENT WITH
APPLICATIONS TO TARGET TRACKING

by

Robert Howard Williams

March 1980

Thesis Advisor: H.A. Titus

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Two Dimensional Matched Filtering for Attitude Measurement With Applications to Target Tracking		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1980
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert Howard Williams		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1980
		13. NUMBER OF PAGES 102
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Two-Dimensions, Matched Filter, Fourier Transform Correlation, Attitude Measurement, Target Tracking

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An application of two-dimensional Fourier Transform correlation for the attitude measurement of three-dimensional objects is considered. Included are geometric optical equations and geometric translation and rotation equations necessary for the image error calculations. Minimum attitude resolution equations are developed to estimate image quantization necessary for a

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601 |

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

given measurement error. These techniques are then applied as an additional measurement to the radar tracking problem.

DD Form 1473
1 Jan 73
S/N 0102-014-6601

Approved for public release; distribution unlimited.

Two Dimensional Matched Filtering for Attitude Measurement with Applications to Target Tracking

Robert Howard Williams
B.S., Oregon State University, 1967

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1980

ABSTRACT

An application of two-dimensional Fourier Transform correlation for the attitude measurement of three-dimensional objects is considered. Included are geometric optical equations and geometric translation and rotation equations necessary for the image error calculations. Minimum attitude resolution equations are developed to estimate image quantization necessary for a given measurement error. These techniques are then applied as an additional measurement to the radar tracking problem.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Continued)

I. INTRODUCTION

The development of sophisticated naval weapons has resulted in the requirement for high accuracy measurement tests and discrete analysis of complex system parameters. Typical of this precise test and evaluation requirement is attitude measurement of three-dimensional airborne objects by processing multiple two-dimensional images.

The most common method used to estimate attitude is by making absolute or precisely relative three-dimensional geometric trajectory measurements of target object extreme structural features (nose, tail, right wing tip, left wing tip, etc.) from camera images. By appropriate camera, range coordinate system, and target coordinate system transformations these measurements will yield an attitude measurement.

A more direct method of attitude measurement involves the correlation, or comparison, of an image of an object at an unknown attitude with a second image of a similar object at a known attitude. At the Naval Weapons Center, an operator controlled film and television correlation assessment technique (FATCAT) facility use this more direct technique for attitude estimation [1].

This thesis presents an enhancement of the correlation/comparison method of attitude measurement by fully automating the technique incorporating a pattern recognition method of the directly digitized image. Image theory, resolution, and filtering leading to the two dimensional matched filter attitude measurement concept is discussed and this concept is applied as an additional measurement to a radar tracking problem.

II. IMAGE THEORY

Three-dimensional to two-dimensional projections are obtained by viewing the real world as recorded by a camera or other imaging device. Image theory defines projection as the transformation of each point in the field of view, through a focal point, and onto a plane. This is also equivalent to locating the focal plane in front of the focal point and changing the sign of the projected points [2]. These geometrical relationships are illustrated in Fig. 1. The transformation of point x, y, z in the X, Y, Z coordinate system of Fig. 1 is given by the following equations:

$$u = \frac{f}{y + f} x \tag{1}$$

$$v = \frac{f}{y + f} z \tag{2}$$

Where u = projected x dimension

v = projected z dimension

If f << y, then the following approximations hold:

$$u = \frac{fx}{y} = kx \tag{3}$$

$$v = \frac{fz}{y} \quad ky \tag{4}$$

Therefore, for a given focal length f, and target range y, a constant image scaling factor k, is calculated.

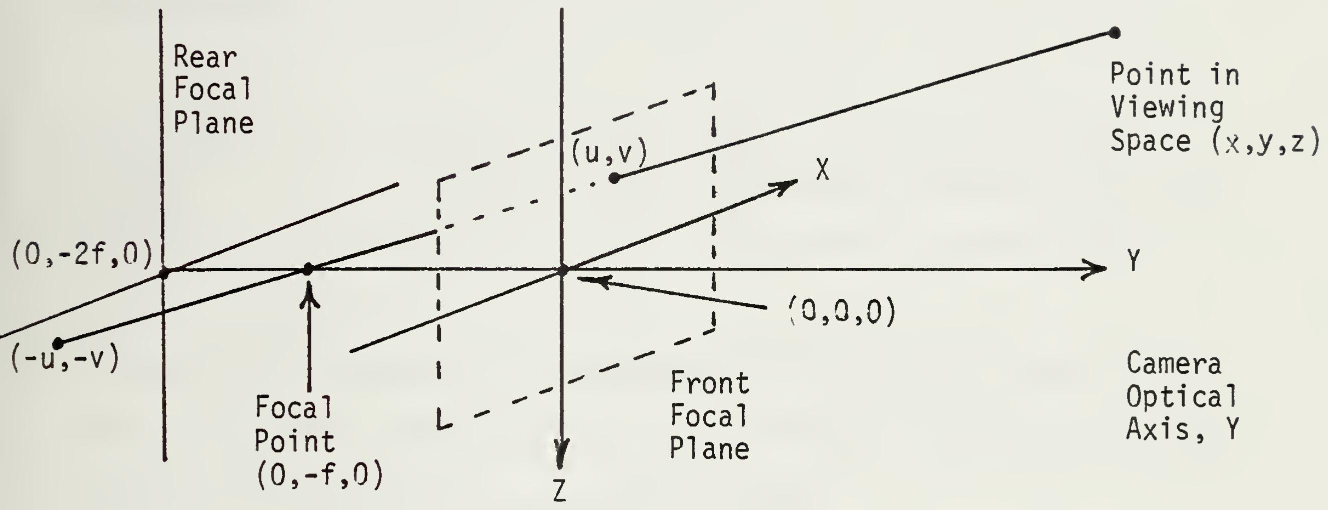

Figure 1

Camera Geometry

A second coordinate system X', Y', Z' can be used to describe a three-dimensional object in the viewing space. If the origins of these two coordinate systems are coincident, and if all the axes are parallel, then the transformation of points from one coordinate system to the other would be simple.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{5}$$

If the axes of the two coordinate systems are rotated by an elevation angle θ around the camera Y axis, by an azimuth angle ψ around the camera Z axis, and by roll angle ϕ around the camera X axis, the transformation equation would be [3], [4]:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (6)$$

If in addition there was a translation of (A,B,C) of the origin of the object coordinate system relative to the origin of the camera coordinate system, the transformation equation would be:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \theta \end{bmatrix} \begin{bmatrix} \psi \end{bmatrix} \begin{bmatrix} \phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} A \\ B \\ C \end{bmatrix} \quad (7)$$

where

$$\begin{bmatrix} \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\begin{bmatrix} \psi \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

III. IMAGE RESOLUTION

The three-dimensional object can be considered to be an aircraft, with maximum orthogonal dimensions on the focal plane of Xmax, Ymax, Zmax. These dimensions are illustrated on the top and side views of Fig. 2. Since the image is to be digitaized, the discrete maximum dimensions can be given in terms of pixel sampling intervals, x and y. A pixel is defined as the smallest unit sampling area (x,y) of resolution for the digitized image.

$$\text{Xmax(quantized)} = XN \tag{8}$$

$$\text{Ymax(quantized)} = YN \tag{9}$$

$$\text{Zmax(quantized)} = ZN \tag{10}$$

These dimensions are illustrated in Fig. 3. The quantization of the image would have a minimum rotation that could be detected by a 1 pixel change at extreme points in the image and would be equivalent to the following angle:

$$\tan\theta = \frac{1}{KN} \tag{11}$$

This relationship is illustrated in Fig. 4. KN would be equal to XN, YN, or ZN depending on the angle of rotation.

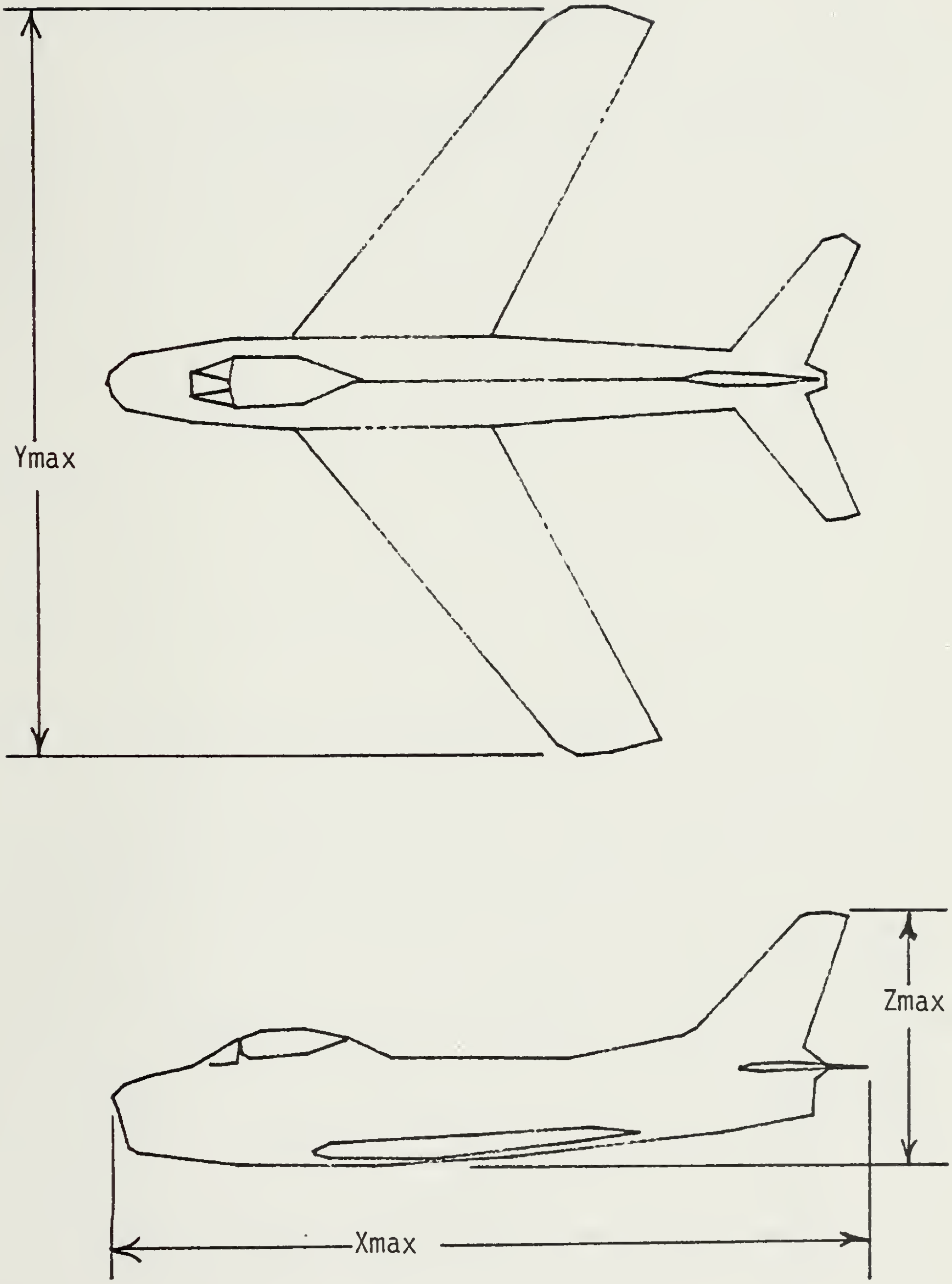

Figure 2

Maximum Image Dimensions, Orthogonal Views

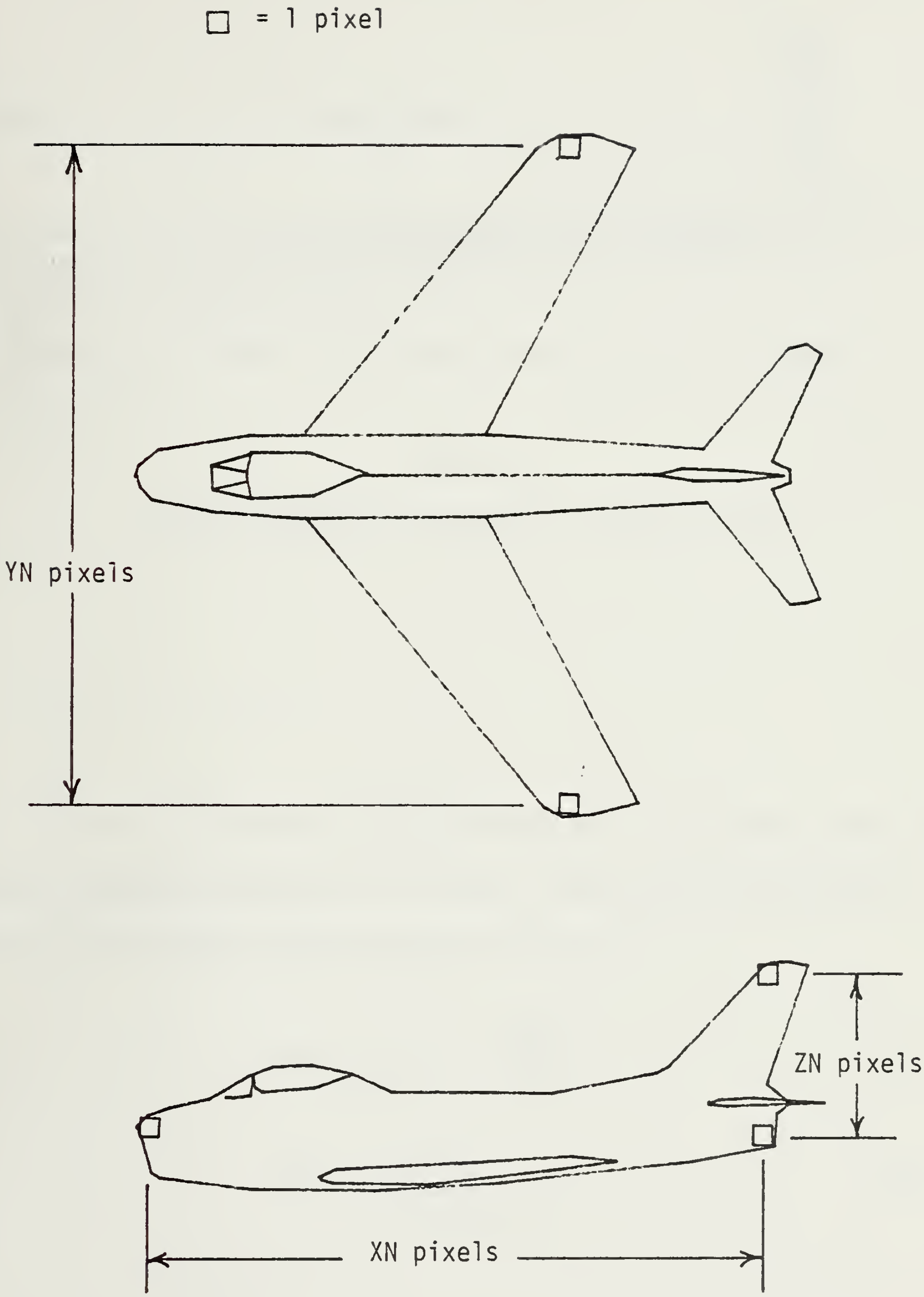

Figure 3

Discrete, Orthogonal Views

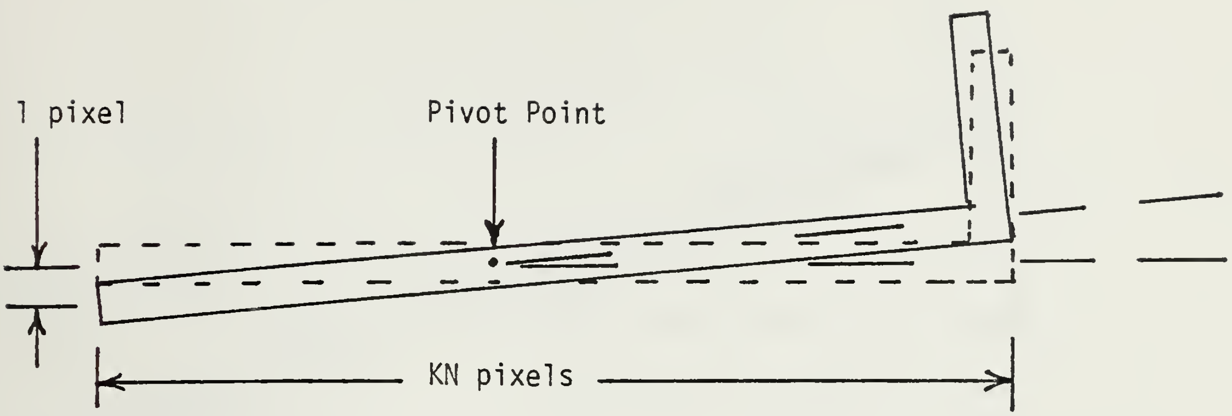

Figure 4

Minimum Detectable Rotation, Orthogonal View

If the axis of rotation was not orthogonal to the focal plane, the minimum detectable rotation of an equal sized object would be much larger, as indicated in the top view of Fig. 5.

$$\cos\theta_{min} = \frac{KN-1}{KN} \tag{12}$$

$$KN = \frac{1}{1-\cos\theta_{min}} \tag{13}$$

Sample values of KN from equations (11) and (13) for minimum detectable rotation are given in Table 1.

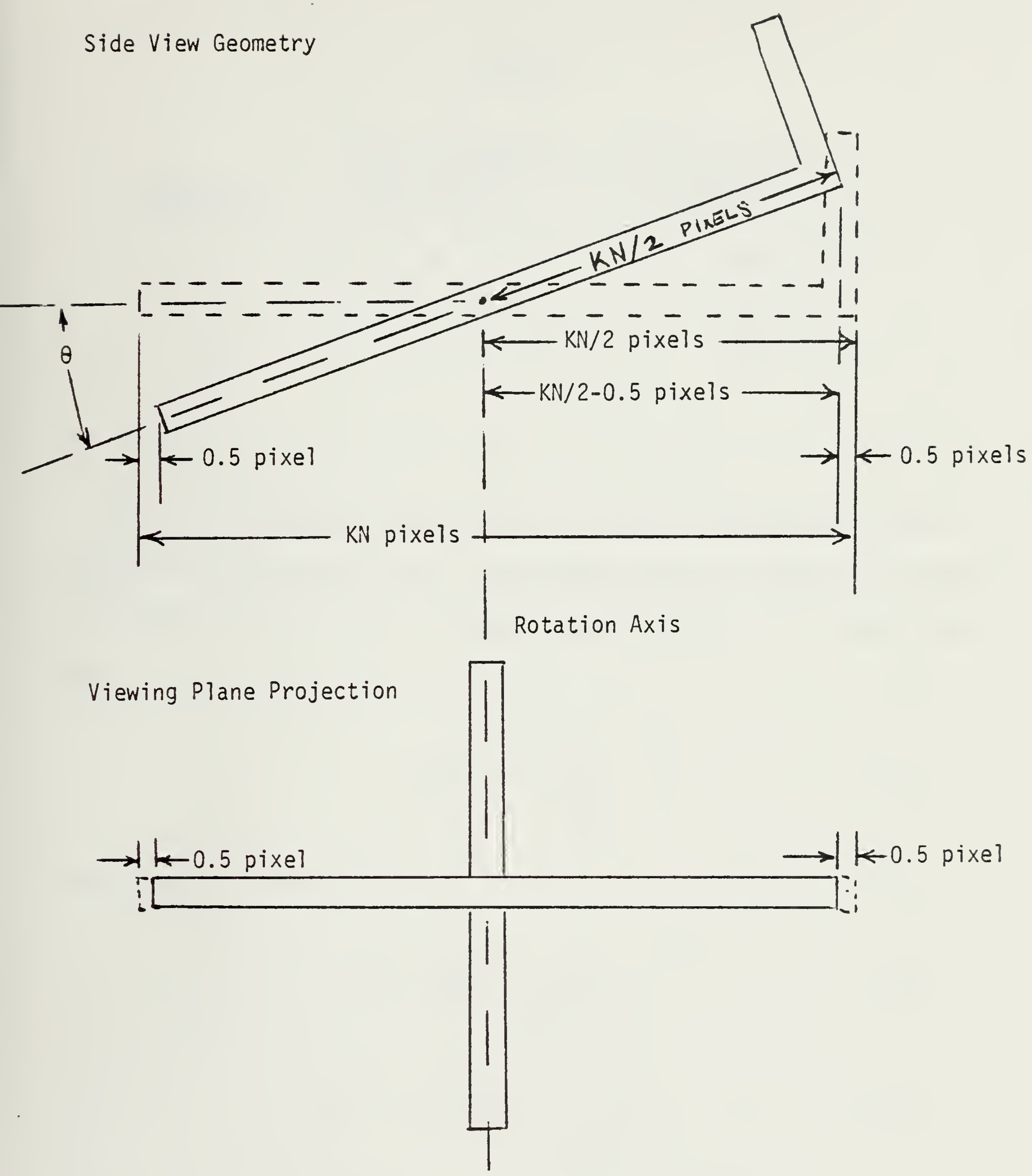

Figure 5. Minimum Detectable Rotation

Rotation Axis in Viewing Plane

TABLE 1

θ (degrees)	Axis Orthogonal to image plane (11) KN (pixels)	Axis Parallel to image plane (13) KN (pixels)
1	58	6566
5	12	263
10	6	66

If it is assumed that the rotational angles are measured by lines through the two extreme pixels, the geometry and error sensitivity would be as shown in Fig. 6. For a maximum length of KN, and a 1 pixel rotation:

$$\tan\theta = \frac{1}{KN} \tag{11}$$

For horizontal errors H:

$$\theta = \arctan \frac{1}{KN \pm H} \tag{14}$$

$$d\theta = d \arctan \frac{1}{KN \pm H} \tag{15}$$

For a change of variables:

$$m = \frac{1}{KN \pm H} \tag{16}$$

$$dm = \frac{dH}{(KN \pm H)^2} \tag{17}$$

$$d\theta = \frac{1}{1 + m^2} dm \tag{18}$$

$$d\theta = \frac{dH}{(KN \pm H)^2+1} \tag{19}$$

For example, if KN = 58, H = 1, then $d\theta/dH$ = 0.0003077 radians/pixel = 0.01763 degrees/pixel. This sensitivity for horizontal errors is very small, as expected.

For vertical errors:

$$\theta = \arctan \frac{1 \pm V}{KN} \tag{20}$$

$$m = \frac{1 \pm V}{KN} \tag{21}$$

$$dm = \frac{dV}{KN} \tag{22}$$

$$d\theta = \frac{dV\ KN}{KN^2 + (1 \pm V)^2} \tag{23}$$

For example, if V = 1 pixel, and KN = 58 pixels, $d\theta/dV$ = 0.01722 radians/pixel = 0.99 degrees/pixel. This is much larger than the horizontal error, as might be expected by observing the geometry of Fig. 6.

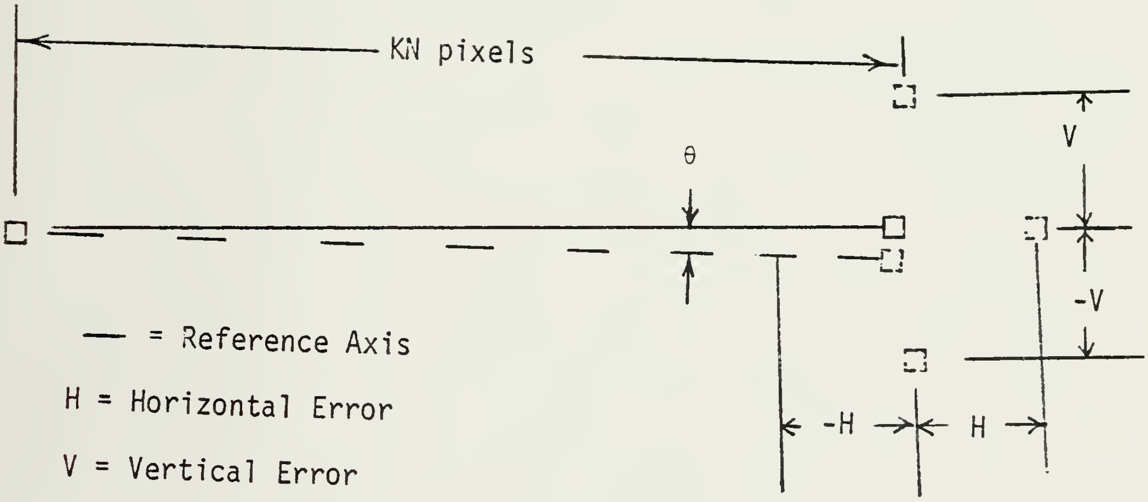

Figure 6.

One-Dimensional Rotation Errors

This section has derived equations (11) and (12) to predict the minimum detectable rotation of a three-dimensional object. These equations apply to a three-dimensional object with a two-dimensional maximum projected dimension of KN pixels. Equation (11) applies if the axis of rotation of the three-dimensional object is parallel with the camera axis. Equation (13) applies if the axis of rotation of the three-dimensional object is orthogonal to the camera axis. Equations (19) and (23) predict the rotational errors resulting from horizontal and vertical measurement errors in extreme points of the two-dimensional image.

IV. IMAGE FILTERING

Two-dimensional matched filtering can be considered to be an extension of one-dimensional correlation filtering [5]. The output of the two-dimensional matched filter will not be a transformed image with some type of improvement or enhancement, but a correlation plane whose amplitude at a given location corresponds to the degree of correlation of the input image (signal + noise) with the desired image (signal). This is equivalent to sliding one photographic transparency across another and adding up each black point that was aligned with another black point on the second image, each white point that was aligned with another white point on the second image, and each grey point that was aligned with another grey point of equal shade on the second image.

Several practical factors must be considered in the actual simulation of the matched filter. The input image and the reference or stored image could both be compared (correlated) as continuous grey scale images, but this would be a tremendous storage and computational task. Therefore, the reference image in the computer simulation is digitized and stored as discrete X,Y,Z points (FACIT Model) in a three-dimensional coordinate system [6]. It is reconstructed, after rotation and translation, as unit amplitude lines projected on the focal plane. It is also assumed that $f \ll y$, and

f and y of equations (1) and (2) are known. This knowledge of focal length and target range yields a constant image scaling factor (k). This type of projection is illustrated in Fig. 7. An example of an aircraft FACIT Model is given in Fig. 8.

The input grey-level image is also transformed into a similar format by use of an edge detector to extract outline information [7]. The edge of the image is defined as the boundary between the two regions of different grey level.

The reference image and the transformed input image are applied to a matched filter to determine the degree of correlation. An ideal correlation function $C(x,y,k,\theta,\psi,\phi)$ would have an impulse response that could be modeled by delta, (sin x/x), triangle or exponential functions. The impulse would occur when the reference and transformed images were:

1. aligned on the image plane (x,y),
2. equally scaled (k) and
3. at the same attitude (θ,ψ,ϕ).

However, due to the presence of noise and the approximation of the three-dimensional object with a finite element model, the correlation response is not delta, (sin x)/x, triangle or exponential in

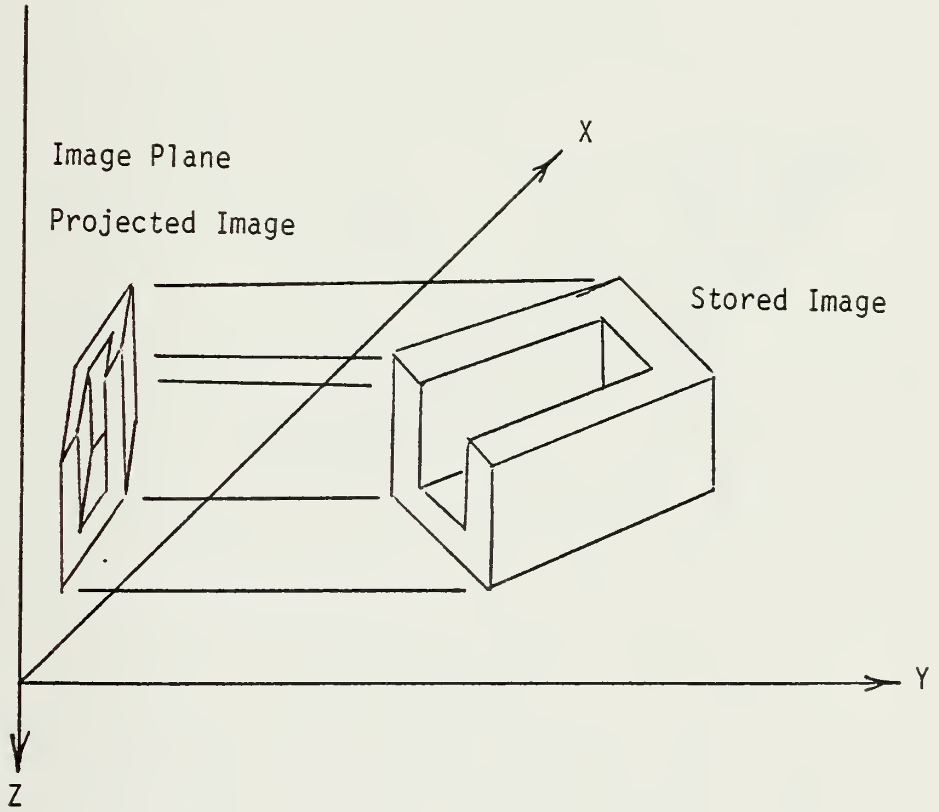

Figure 7.

Three-Dimension to Two-Dimension Projection

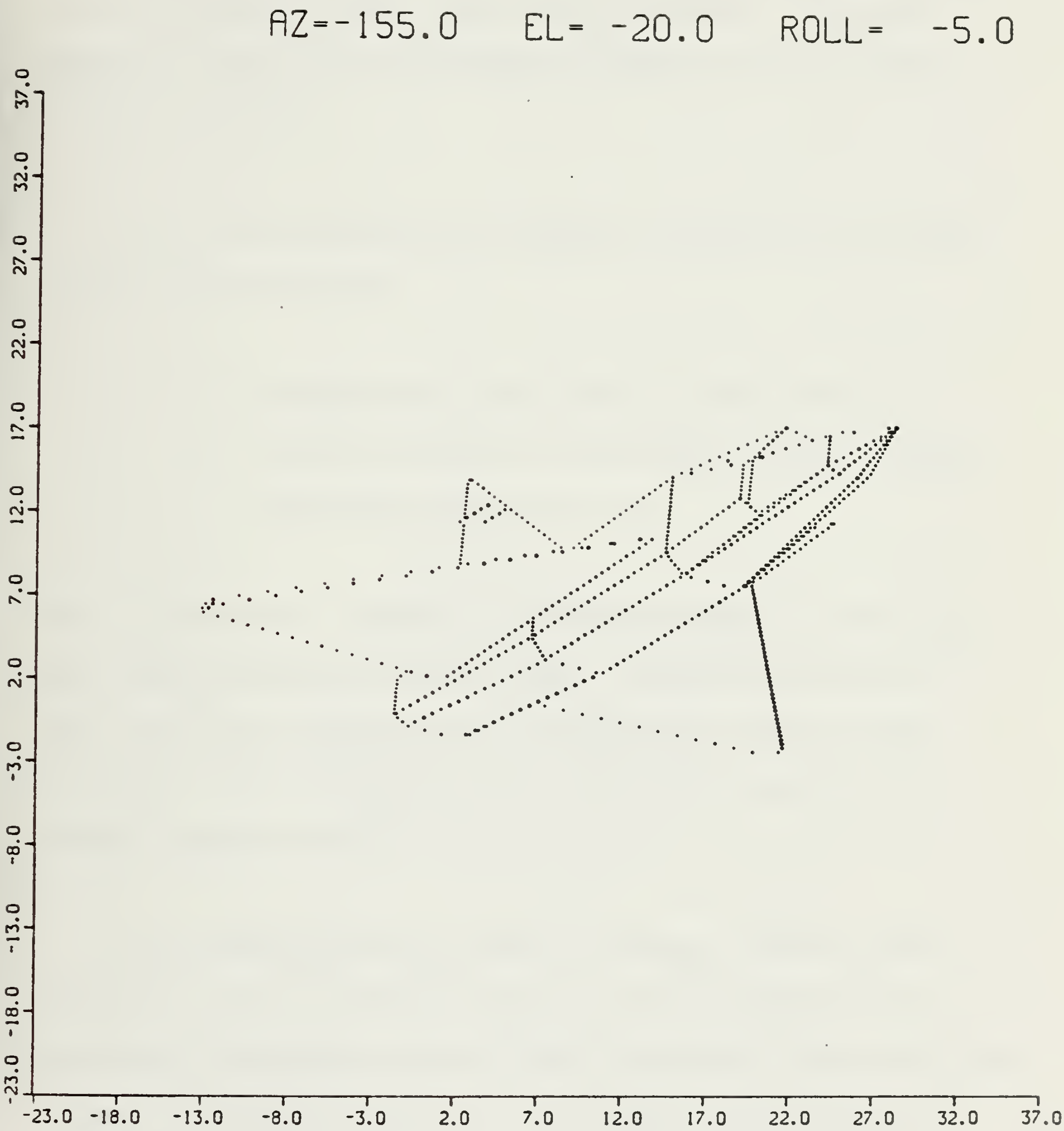

Figure 8. FACIT Model of F-102

shape. It is not necessary to know the exact correlation response function or even the constant component due to a nonzero mean value of the reference and input image variable function. It is only necessary to know the values of the variables x, y, k, θ, ψ, and ϕ at peak correlation.

The following simplifying assumptions are made for an attitude (θ,ψ,ϕ) only correlation:

1. the images have been scaled to an equal size (k)
2. the peak correlation on the image plane will indicate translational alignment (x,y)

Fig. 9 illustrates a typical x-y correlation between a noisy image and computer-generated reference image. The peak correlation at (0.0, 0.0) pixels indicates that both images are aligned in the x-y plane. This is because the noisy image was generated from a computer reference image.

The attitude of the input image is unknown. Therefore, the measurement of the attitude of the input image is equivalent to applying the transformed image to a bank of matched filters with each "tuned" to a specific attitude. This process is illustrated in Fig. 10.

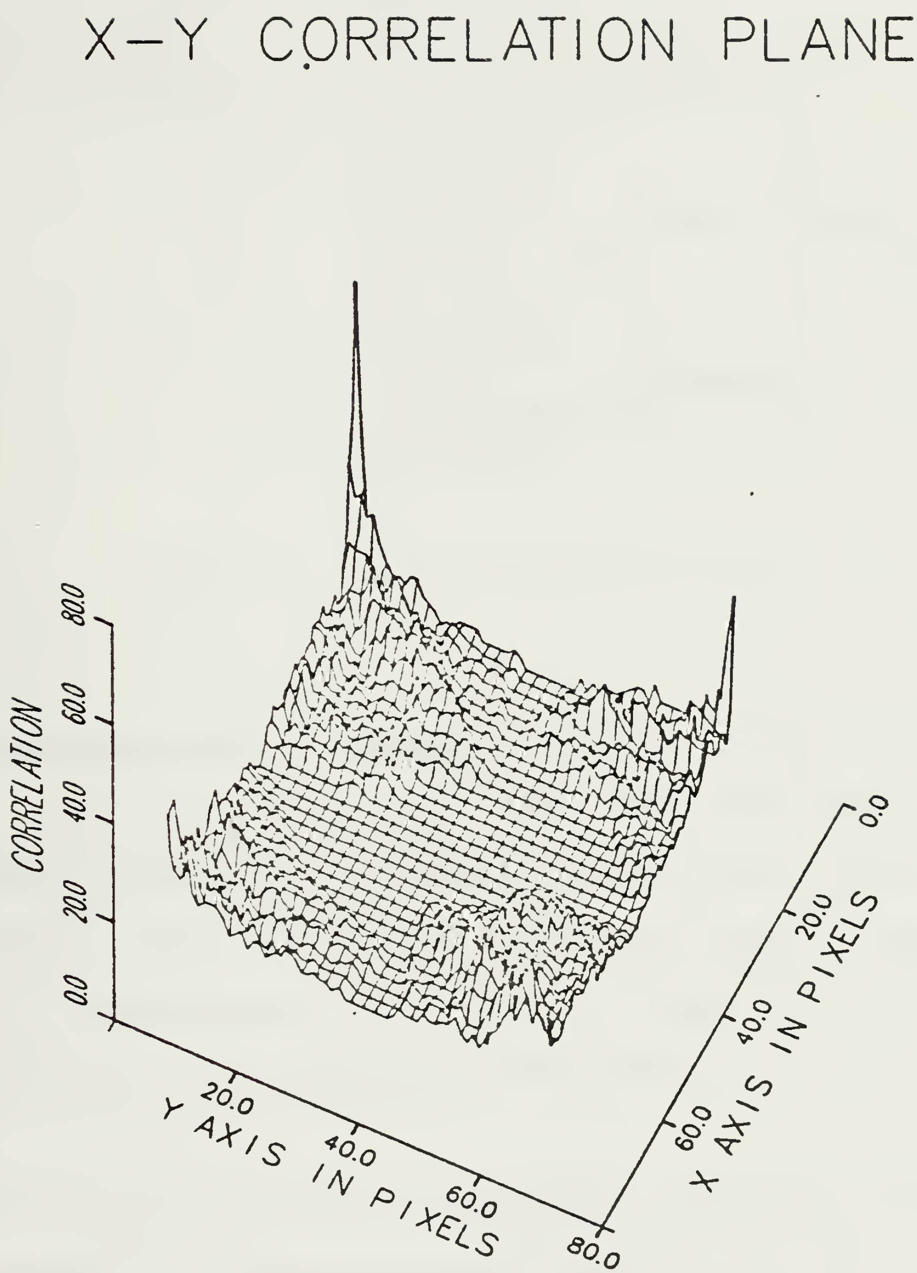

Figure 9

Sample 64 X 64 Pixel X-Y Correlation Plane Obtained from 32 X 32 Pixel Image and 32 X 32 Pixel Reference

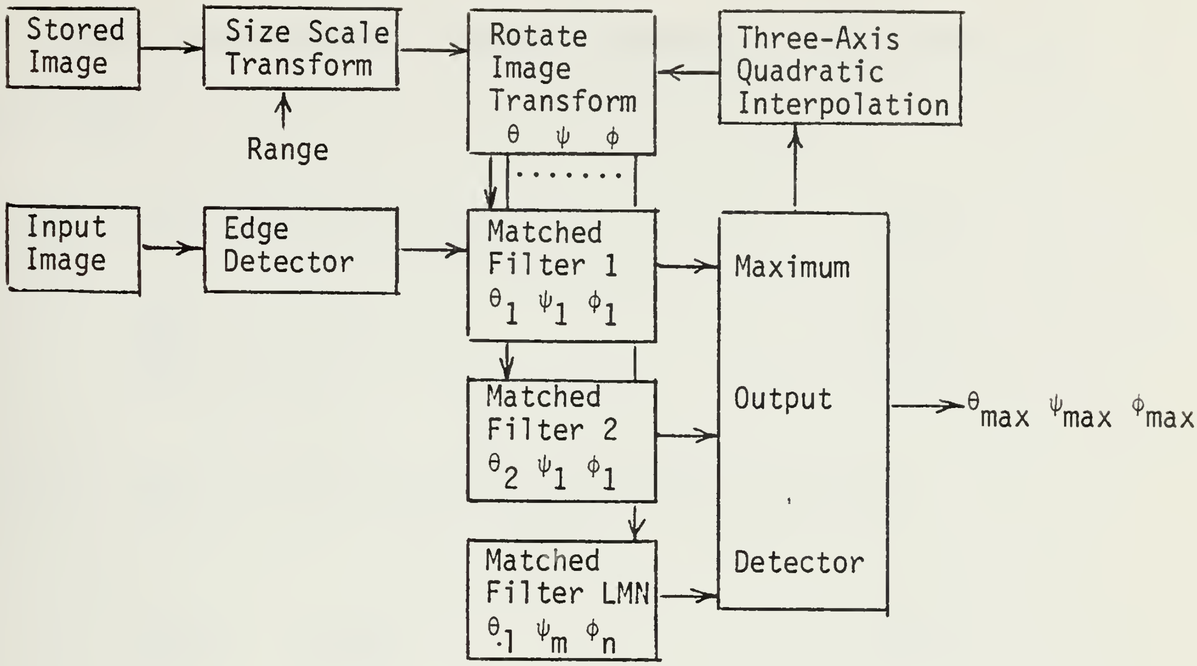

Figure 10. Image Processing Flow Diagram

If there was a priori knowledge of azimuth θ, elevation ψ, and roll ϕ, to within 10 degrees each, and a 1 degree resolution of measurement was required, a brute force approach would require $10 \times 10 \times 10 = 1000$ correlations or 1000 matched filters. Three axis quadratic interpolation is used to reduce the number of correlations required for the given resolution of measurement [8].

It is assumed that the variables azimuth, elevation, and roll are uncoupled in the joint correlation function $C(x,y,\theta,\psi,\phi)$. Assuming that there is a near orthogonal relationship allows $C(x,y,\theta,\psi,\phi)$ to be sequentially maximized (minimized) with the independent variables: x, y, azimuth, elevation, and roll.

The pitch correlation function is modeled as a quadratic function $C(\theta)$:

$$C(\theta) = a\,\theta^2 + b\,\theta + c \qquad (24)$$

$$\frac{dC}{d\theta} = 2\,a\,\theta + b \qquad (25)$$

Equation (25) is set to zero to maximize (minimize) $C(\theta)$.

$$\theta_{max} = \frac{b}{2\,a} \qquad (26)$$

$\theta_1 = 0$ degrees, $\theta_2 = \theta_1 + \Delta\theta$ degrees, and $\theta_3 = \theta_1 - \Delta\theta$ degrees are substituted in equation (24) to solve for the unknowns a, b, and c. The solution of equation (26) is:

$$\theta_{max} = \frac{(C(\theta_3) - C(\theta_2))\,\Delta\theta}{2C(\theta_2) + 2C(\theta_3) - 4C(\theta_1)} \qquad (27)$$

Similar equations are written for $C(\psi)$, and $C(\phi)$.

The computer simulations listed in Appendix B implement equaion (27). The three correlations $C(\theta_1)$, $C(\theta_2)$ and $C(\theta_3)$ are calculated with $\Delta\theta = 5$ degrees for a rough approximation of θ_{max}. Then three

correlations $C(\theta_1)$, $C(\theta_2)$ and $C(\theta_3)$ are calculated with $\Delta\theta = 2$ degrees for a sharper quadratic approximation of θ_{max}. In addition, the ± 2 and ± 5 degree intervals are tested to make sure that the mid-point correlation $C(\theta_1)$ was greater than or equal to the end point correlations $C(\theta_2)$ and $C(\theta_3)$. If either is not, the interval is shifted by 2 or 5 degrees in the direction of the larger correlation and the mid-point is tested again. This dual pass computation reduces the unwanted coupling effect of the attitude variables. The ± 5 degree calculation detects the general angle of maximum correlation. The ± 2 degree calculation reduces the estimated angle error by not being as sensitive to unsymmetry and nonlinearities in the correlation function.

A sample of the dual pass azimuth estimation is illustrated in Fig. 11. For this sample, noisy edge data of a simulated aircraft at an azimuth angle of -45.0 degrees is used as input data. This data is correlated with computer-generated edge data from a stored aircraft model. The initial estimate of attitude is -48.0 degrees. The final azimuth calculated is -44.11 degrees, a 0.89 degree error from the true -45.0 degree aircraft azimuth attitude. This is a typical maximum error. For example, if the initial estimate of azimuth is -42.0 degrees, the final error would be 0.0 degrees.

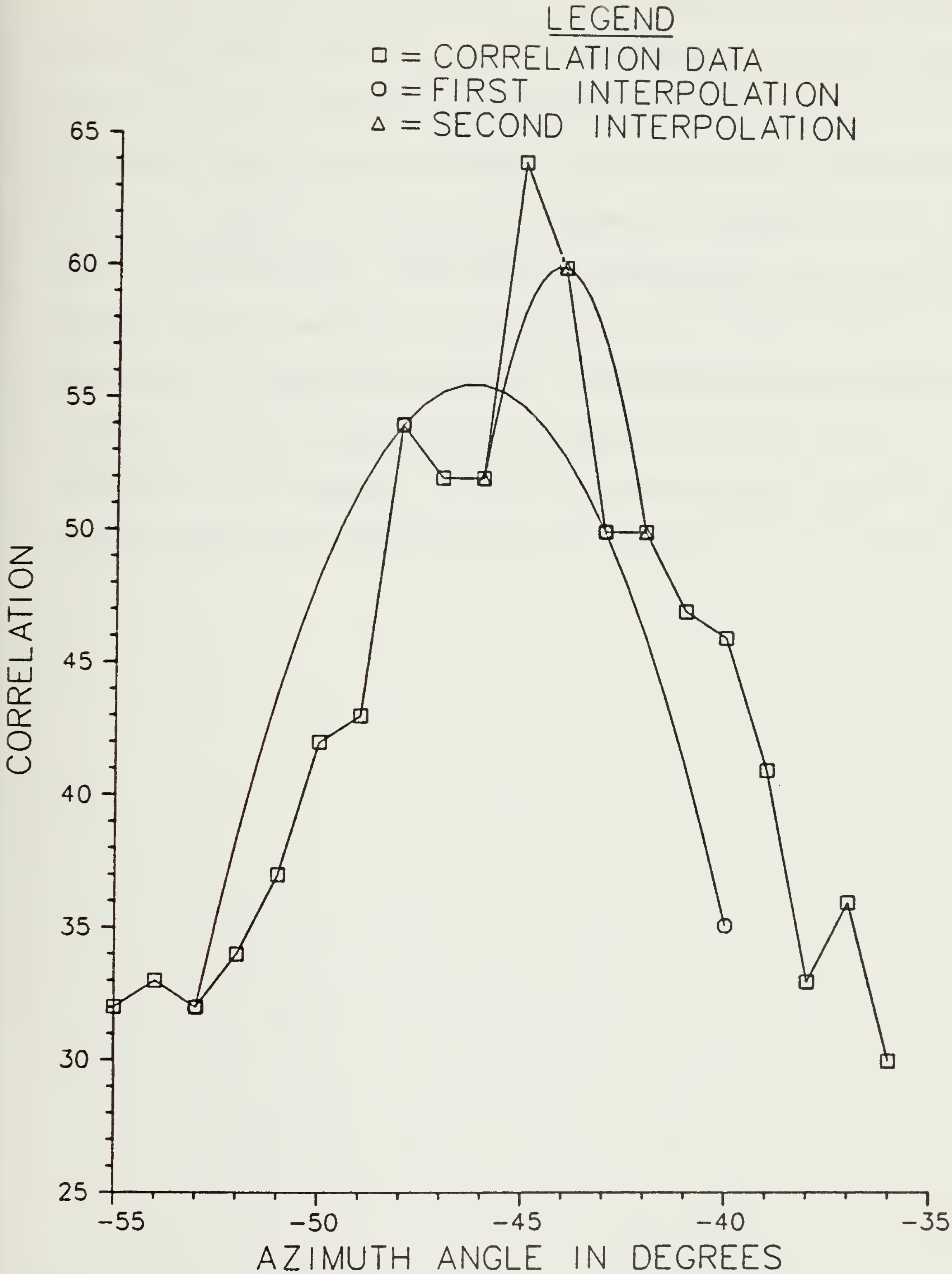

Figure 11

Azimuth Correlation Function

This section has introduced the use of two-dimensional correlation to determine the attitude of a three-dimensional object. Equation (24) is used as a second order polynomial model for the correlation function of each attitude variable. This approximation reduces the number of computations necessary to determine the peak correlation for each attitude variable. This quadratic approximation was demonstrated with a computer simulation that tested for the maximum correlation of an image as a function of attitude. The simulated image was previously generated by adding edge noise to a computer-generated image at a known attitude of -45.0 degrees. By starting the correlation algorithm with a -3.0 degree offset, the final error was found to be 0.89 degrees.

V. MATCHED FILTER EQUATION

It is well known that one-dimension correlation can be calculated by a simple multiplication of two functions that are Fourier transformed [9].

$$c(t) = \int_{-\infty}^{\infty} x\,(\tau)\; h(t+\tau)\; d\tau \;=\; x(t) * h(-t) \tag{28}$$

where τ is the dummy variable of integration and * is used to symbolize the convolution integral.

$$C(w) = X(w)\; H^*(w) = X^*(w)\; H(w) \tag{29}$$

where $w = 2\pi f$ = frequency in radians and * is used to indicate complex conjugation; that is, if

$$H(w) = R(w) + jI(w), \text{ then } H^*(w) = R(w) - jI(w).$$

$$c(t) \quad \Longleftrightarrow \quad C(w) \tag{30}$$

where <=> is used to indicate a Fourier transform pair; that is

C(w) is the Fourier transform of c(t).

This same relationship can be shown in two-dimensions and in discrete form [5].

$$c(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} x(i,j)\, h(k+i, l+j) \qquad (31)$$

$$C(m,n) = X^*(m,n)\, H(m,n) \qquad (32)$$

$$C(m,n) = \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} x(k,l) e^{j \frac{2\Pi mk}{K}} e^{j \frac{2\Pi nl}{L}} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} h(i,j) e^{j \frac{2\Pi mi}{I}} e^{j \frac{2\Pi nj}{J}} \qquad (33)$$

m = variable in first dimension

n = variable in second dimension

K = I = M = Period in first dimension

L = J = N = Period in second dimension

A complete listing of the transformation indicated in equations (32) and (33) is given in Appendix A.

The reason the Fourier domain correlation approach is used is that for large data arrays the Fast Fourier Transform (FFT) computation time is less than that of direct time or space domain correlation. Also, Fourier transform and array processing hardware is available for many scientific computers.

The continuous correlation equation (28) states that an infinite integration is necessary to determine the correlation at a particular time. This is not the case for finite sampled functions. The summation interval for the sampled functions requires that the majority of the information in the two functions be contained in the finite sampled interval for the correlation to approximate the continuous correlation. When x(t) and h(t) are sampled with an impulse function $\Delta(t)$ over a finite interval, $0 < t < t_{max} = NT$, their Fourier transforms $x(w)*\Delta(w)$ and $H(w)*\Delta(w)$ will be periodic with period $w = 1/T$, where T = sampling interval of the impulse function $\Delta(t)$. When the continuous function x(t) is sampled over the interval $0 < t < t_{max}$ and Fourier transformed, this is equivalent to calculating the Fourier Series coefficients of a sampled periodic sequence $x_p(nT)$ of period N, where $N = t_p/T$ and T = time interval between samples. If $x_p(nT)$ is time shifted in relation to x(nT) by m samples, and the first N samples are transformed, it can be shown that the resulting complex Fourier transform

will change in phase but not in amplitude or period [9]. In the time domain, this shift of m samples is due to the periodicity of x_p (nT) and is referred to as a circular shift of a sequence [5]. The sequence appears to be circular because the last sample of x(nT) that is shifted beyond the interval N T appears at the beginning of the sequence. Likewise, in the two-dimensional correlation there is a circular shift in two dimensions when one image x(i,j) is correlated with another h(i,j). The two-dimensional periodic or circular sampled function is equivalent to joining parallel horizontal and vertical edges of each image and rotating one image on each of its axes while the other image remains stationary. This two-dimensional circular shifting is depicted in Fig. 12. In this figure, image 2 has shifted n samples in the horizontal direction with respect to the fixed image 1. Image 2 has shifted m samples in the vertical direction with respect to the fixed image 1. To clarify that image 2 has circularly "wrapped around" image 1, the corners are labeled A, B, C and D.

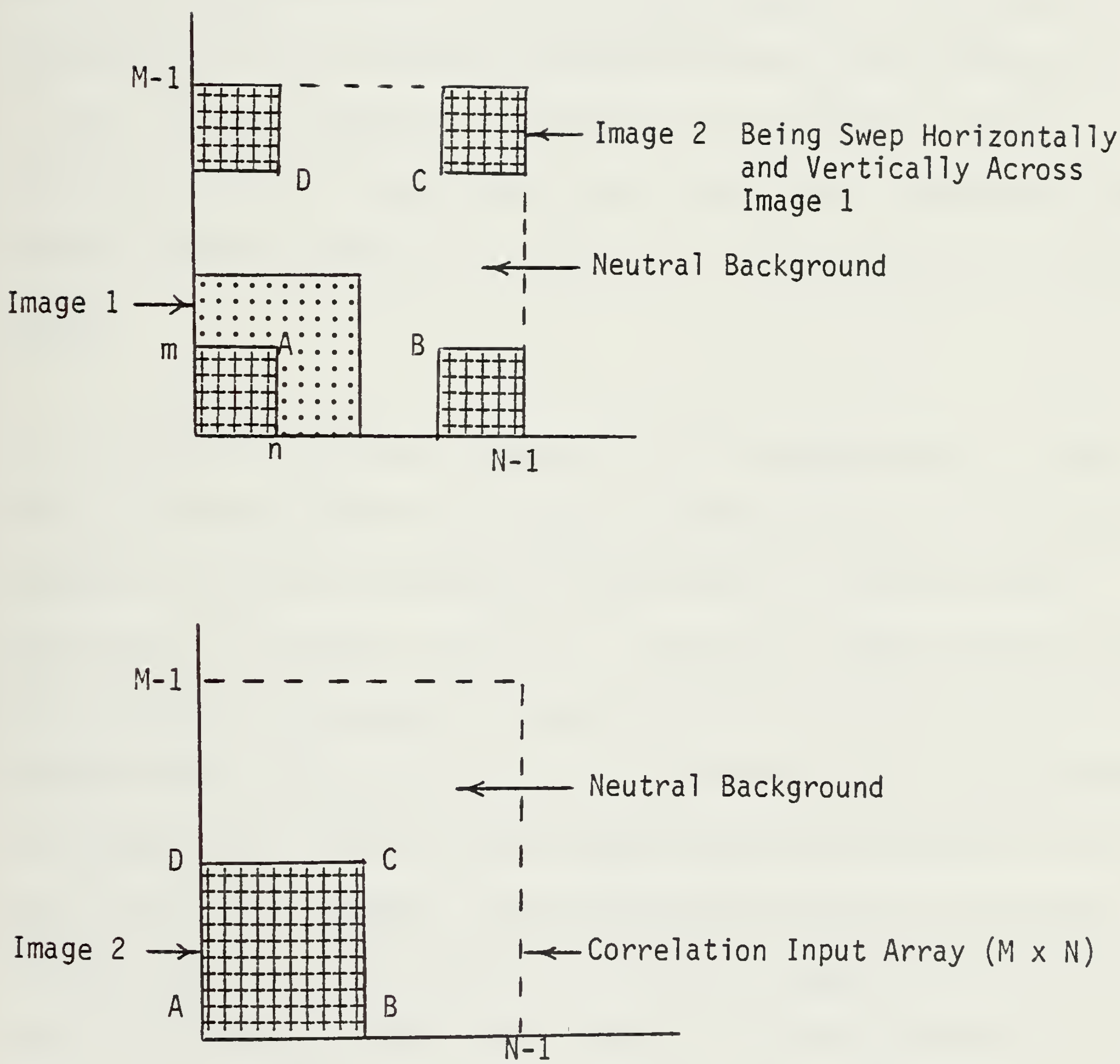

Figure 12. Implementation of Two-Dimensional Correlation

The continuous images 1 and 2 are sampled at N and M discrete points in the x and y axis respectively. This sampling results in a periodic equation (33) in the transformed frequency domain of period N and M. If the correlation product of the two sampled images has nonzero terms beyond the N and M dimensions, there will be an overlap of nonzero samples in the frequency domain. This overlap phenomenon of a high frequency component taking on the identity of a lower frequency component is referred to as frequency aliasing. Likewise, spatial aliasing would occur if there was an overlap of the periodic spatial functions; for example, if the first and last samples added.

To test the correlation method of attitude measurement, a computer simulation program, FAC listed in Appendix B, was run. Discrete boundary points were generated as input data for a known aircraft attitude as illustrated in Fig. 13. A noisy test object was generated that was within ± 5 degrees in azimuth, elevation and roll of the reference object. The noisy boundary, as described in the next paragraph, was compared with the noise-free computer-generated edge data to estimate the aircraft attitude. The representative errors of attitude (azimuth, elevation and roll) were computed by comparing the attitude that was iteratively predicted by equation (27) with the actual attitude. These results are summarized in Table II.

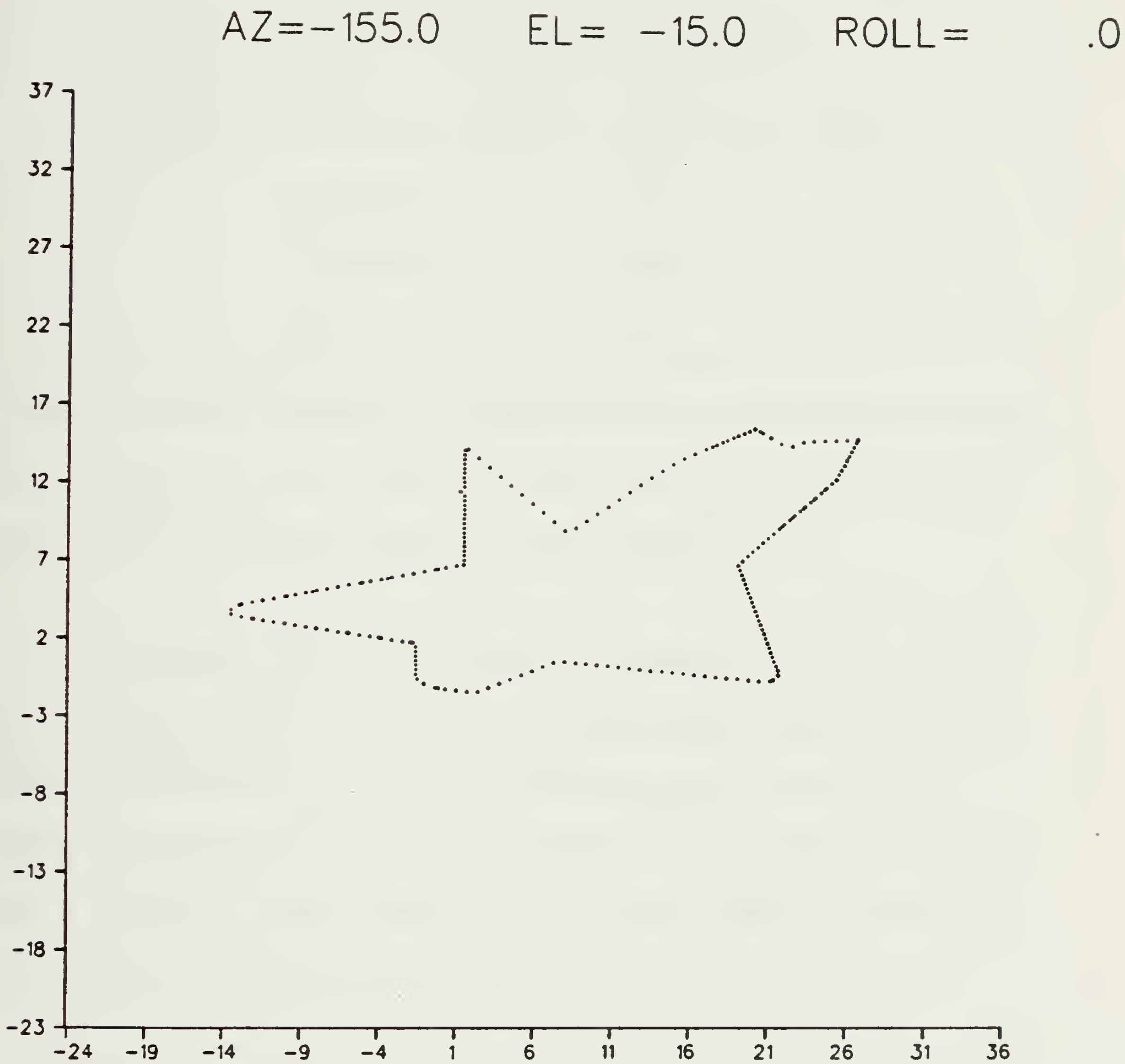

Figure 13

Boundary Points of FACIT Model

Table II.

Representative Attitude Errors (Degrees) Three Sample Average, 1 Pixel Noise Standard Deviation	
Azimuth	0.12
Elevation	0.55
Roll	0.71

It was noted that the errors increased with each attitude measurement. This is because of the sequential coupling of a previous attitude angle error to the next quadratic angle interpolation.

The quadratic correlation attitude estimator (27) was initialized with data to within 10 degrees of the known aircraft attitude. Boundary points on a 32 x 32 grid were generated from the two-dimensional projection of the test aircraft. The number of boundary points generated were a function of the test aircraft attitude, but typically 108 points were generated:

(27 each: maximum horizontal, minimum horizontal, maximum vertical, minimum vertical).

To simulate detector noise, Gaussian noise with zero mean and one pixel standard deviation was added to each boundary point, horizontally and vertically. Since the boundary point function was on a discrete 32 x 32 grid, this had the effect of quantizing the noise to unit pixel intervals. For example, if the absolute value of the Gaussian noise was less than 0.5 pixel, no noise would be added. If the absolute value of the Gaussian noise was between 0.5 and 1.5 pixels, one pixel of noise was added. Likewise, two or more pixels of noise would be added for greater outputs of the generator. These noisy boundary points are shown in Fig. 14.

An uncalibrated system test with real data was made from a digitized video image of a F102 aircraft. The intensity data from the image was transformed into edge points. The edge points were correlated with the model data stored in the computer memory. This predicted the aircraft's attitude.

The x,y resolution of the digitization was 340 pixel columns x 220 pixel rows and the intensity resolution was 1 to 256. As the photograph in Fig. 15 shows, the aircraft image makes up only a small portion of the video image. Therefore, a 64 x 64 pixel window shown in Fig. 16 was contrast enhanced, in that the original intensity range of 103 to 156 was linearly transformed to a 1 to 256 intensity level. It is noted that the digitized image is not displayed at the

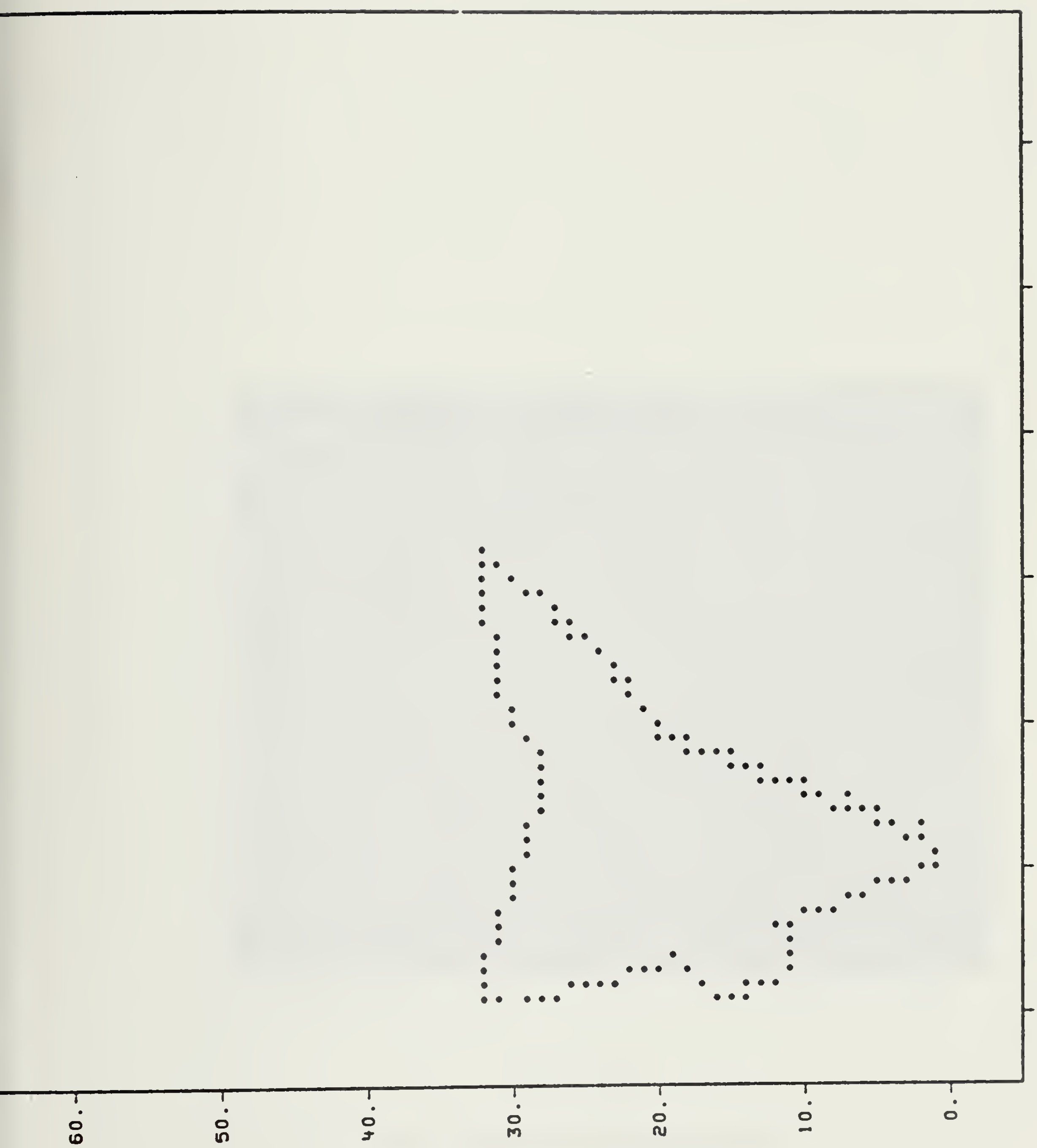

Figure 14. Boundary Points of FACIT Model with Added Noise

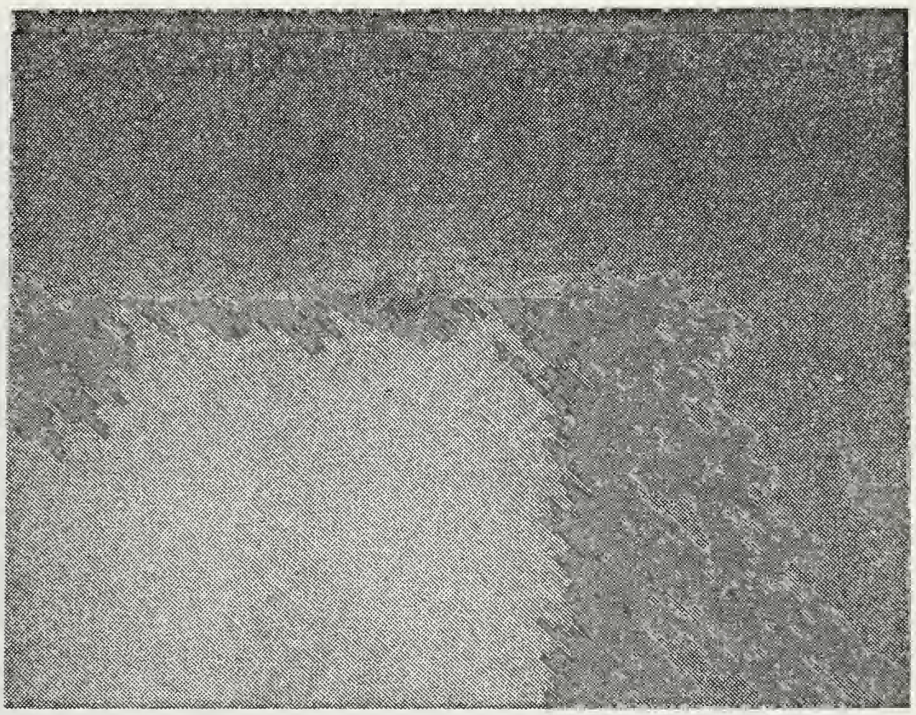

Figure 15

Photograph of F102 Aircraft

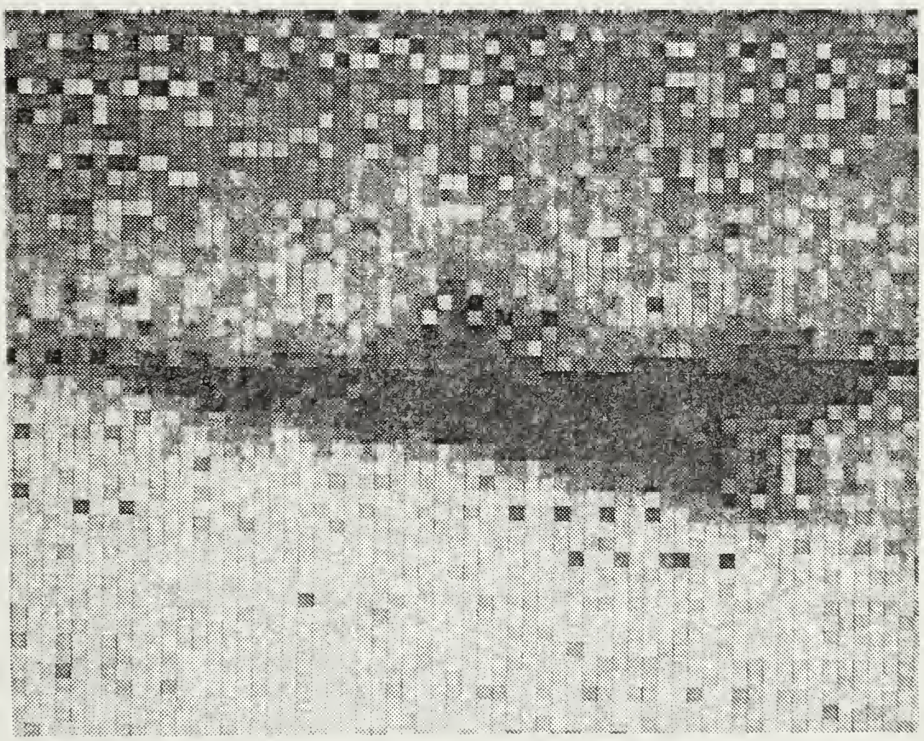

Figure 16

Digitized 64 X 64 Pixel Image of F102 Aircraft

correct aspect ratio due to the unequal sampling ratios taken vertically and horizontally.

The edge data from Rosenfeld's Detector [7] was applied to a variable threshold iteration that was adjusted until greater than 200 pixels were above the threshold. This adjustment was necessary so that low contrast edge pixels typically at wing, nose and tail extremities would exceed the threshold. These extremities are most important in obtaining a given attitude accuracy. Fig. 17 shows the results of the best computer-generated match of the stored image superimposed on the edge data obtained from the video image. The computer predicted the attitude to be:

Azimuth	=	-115.0 degrees
Elevation	=	45.0 degrees
Roll	=	- 5.0 degrees

There was no other data available to establish the attitude of the F102 aircraft at the instant the camera recorded the image. It is presumed that further investigation would show this attitude to be correct to within the limits of the image resolution and the edge noise present.

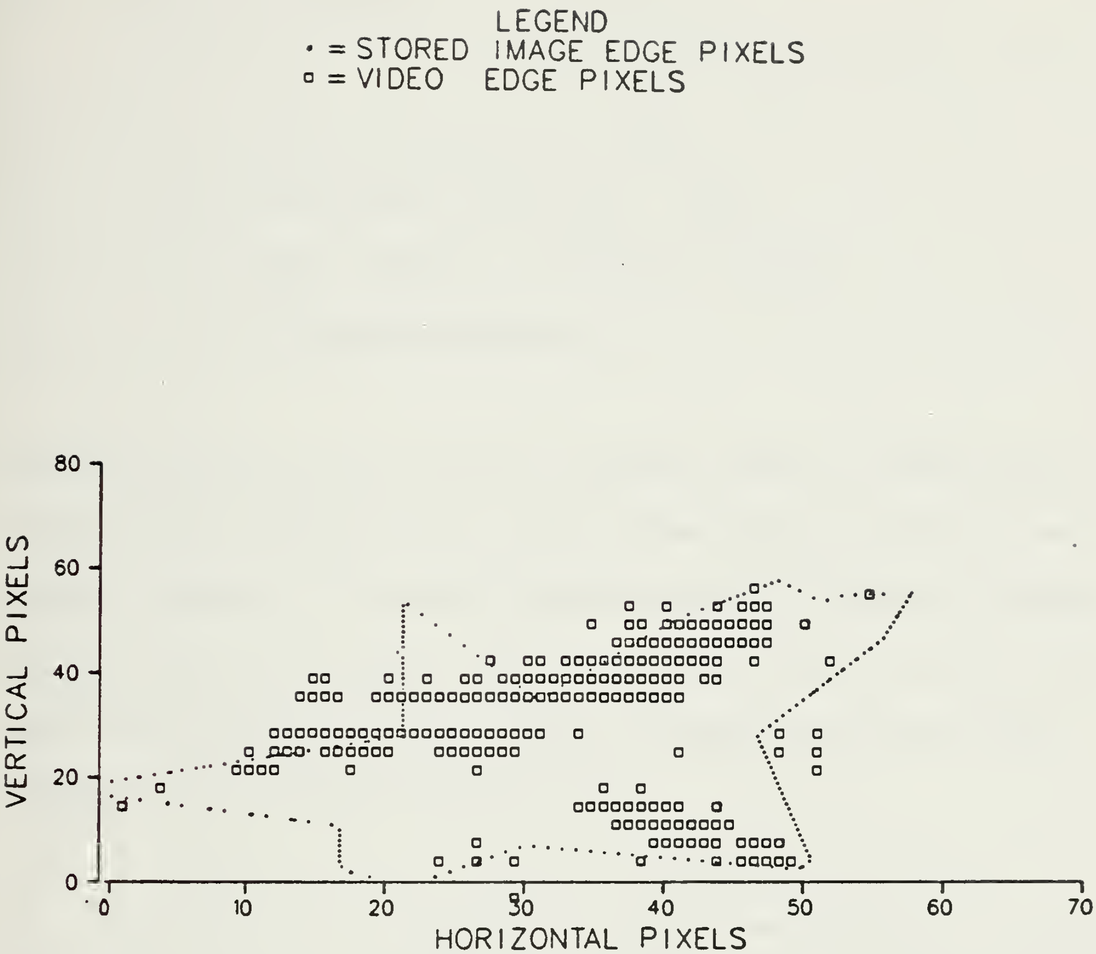

Figure 17

Best Fit of Stored Image Pixels to Video Outline Pixels of Fig. 16

The lack of sharpness in the transformed edge data shown in Fig. 17 can be attributed to several factors;

1. Low x,y resolution (64 x 64).
2. Low intensity range (103 - 156).
3. Uneven lighting and shading on edges.
4. A simple edge detector.

The last factor would be the only one that could be significantly changed given a free-flying aircraft, fixed camera locations, and natural lighting. A review of methods used for extracting features based on edges and contours is included in a survey by Lavine [11]. More recent theories and experiments on edge detectors have been demonstrated by Abdou and Pratt [12].

This section has introduced the use of two-dimensional Fourier domain correlation as described by equation (32). Representative results of three computer correlation simulations were listed. The simulation compared a computer generated edge model to a previous computer generated model with added edge noise. Finally, a sample correlation of the computer generated edge model with the edge data detected from a real video image of an aircraft at an unknown attitude was described.

VI. AIDED AIRCRAFT TRACKING

An example of the utility of the attitude measurement is demonstrated by utilizing an aircraft positional estimating Kalman Filter with combined radar and electro-optic sensor measurements [10]. It is difficult for the radar-only estimator to keep a close target in track if it is highly maneuverable. The criterion of "highly maneuverable" for a particular radar is dependent on the following:

1. The sampling interval of the radar.
2. The range gate period, or the sampling aperture.
3. The receiver sensitivity.
4. The return signal to noise ratio.
5. The mount azimuth and elevation dynamics, or time constants.
6. The measurement accuracy of azimuth, elevation and range.
7. The point source tracking of a finite length target.

These parameters and others contribute to the ability of the radar to track with minimum error. By modeling the aircraft target to be driven by an acceleration function that is dependent upon aircraft attitude, more accurate range, range rate and range acceleration state estimates are possible. The acceleration function is based on the well-known aerodynamic coupling between the target attitude and the direction of the target acceleration. This coupling is summarized in the following comments:

1. Major accelerations (lift) are normal to the velocity vector and the pitch axis.
2. Positive lift is more likely than negative lift due to pilot physiological factors and to structural loading design.
3. Accelerations in the velocity direction (drag/thrust) are generally smaller in magnitude and of shorter duration than the lift accelerations.

A complete multiple measurement system using the previously discussed imaging attitude measurement concept as a radar tracking aid to the positional estimating Kalman Filter is shown in Fig. 18. In the upper left corner of Fig. 18 there are n radars that provide range, azimuth and elevation measurements to the Iterated Extended Kalman target estimator. In the lower left corner there are m video cameras that input data to m correlators. These correlators first scale the stored model to a size equal to the predicted input image size. The scaling is a function of target range and system gains such as telescope magnification and image pixel sampling area. The correlators then apply equation (33) as described in the previous section for the computer simulation FAC. The attitude measurements azimuth, elevation, and roll, from the correlators, are transformed into a common coordinate system. The predicted attitude of a model aircraft in a coordinated turn is combined with the m attitude measurements in a Kalman Filter.

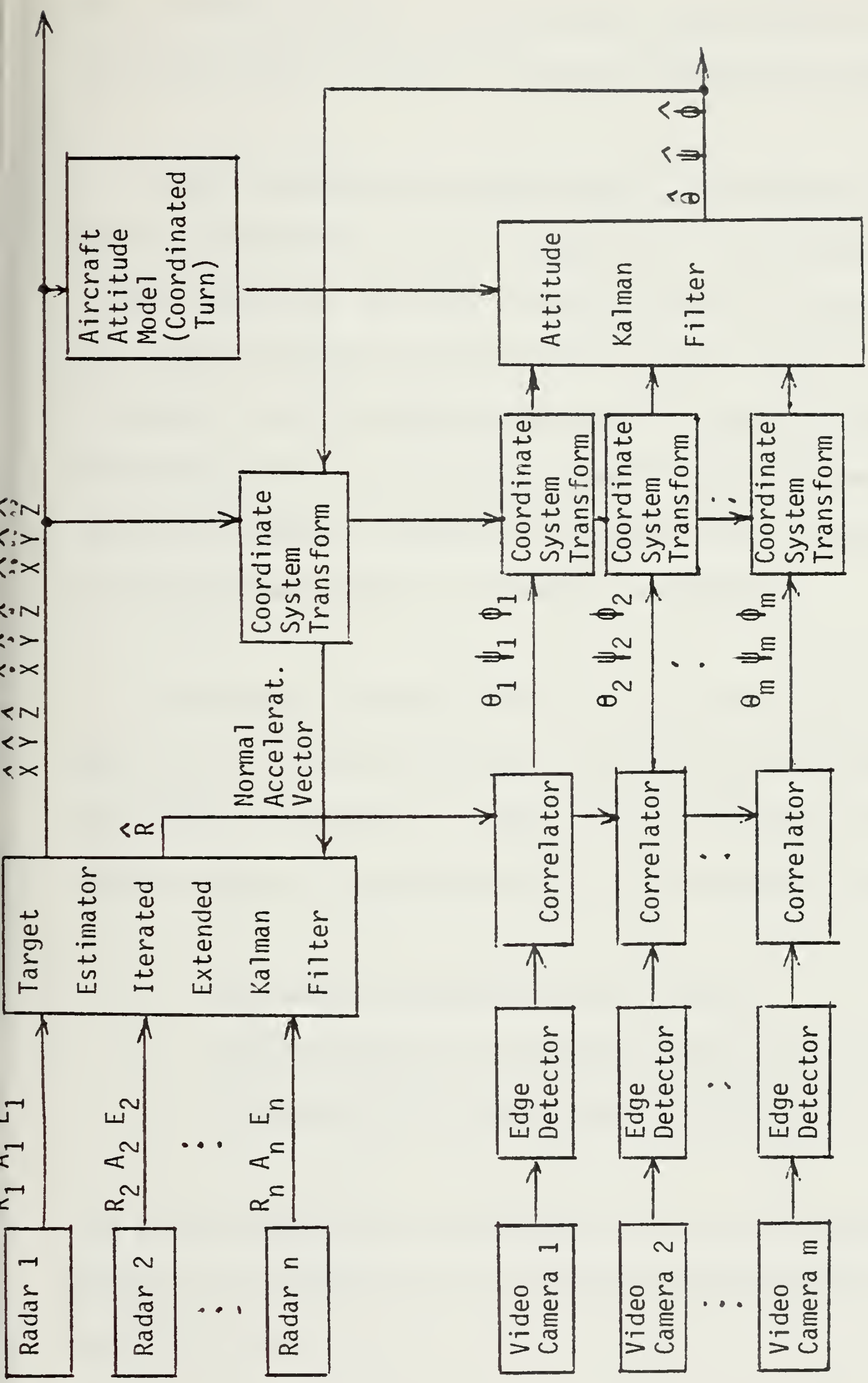

Figure 18. Attitude Measurement and Tracking System

The estimated attitude from the Kalman Filter predicts an acceleration vector that is used as an input to the target estimator.

The simulation of a target range (R) estimation was performed with the program KAL listed in Appendix B. The simulation KAL was a limited example of the radar tracking filter as illustrated in Fig. 18. This simplification was made possible by not including some tracking variables, such as azimuth and elevation, to the full three-dimensional trajectory filter. The trajectory was truly three-dimensional, hence there were changes in azimuth and elevation, but they were not used as system measurements or estimated as parameters.

The Iterated Extended Kalman Filtering subroutine FTKALI is a modification of the IMSL Kalman Filtering subroutine FKALM [14]. The system was modeled as a radar track of a nominal constant acceleration target. The assumptions for this trajectory are:

1. The average acceleration rate is zero.
2. The acceleration rate between radar scans is constant.
3. The acceleration between scanning intervals in uncorrelated.

The above assumptions are described in the well-known Alpha-Beta-Gamma Filter equations [13]. Written in a discrete form, these equations are:

$$x_n = x_{n-1} + T\,\dot{x}_{n-1} + \frac{T^2}{2}\ddot{x}_{n-1} + \frac{T^3}{6}\,ar_{n-1} \tag{34}$$

$$\dot{x}_n = \dot{x}_{n-1} + T\,\ddot{x}_{n-1} + \frac{T^2}{2}\,ar_{n-1} \tag{35}$$

$$\ddot{x}_n = \ddot{x}_{n-1} + T\,ar_{n-1} \tag{36}$$

where ar is a random sequence with zero-mean and a mean-square acceleration rate of

$$E\left[ar_j \quad ar_k\right] = \begin{bmatrix} \sigma^2 & \text{for } j = k \\ 0 & \text{otherwise} \end{bmatrix} \tag{37}$$

The target trajectory was simulated in three parts:

1. A target constant closing velocity of 100 or 175 yards/second at the same altitude as the radar.
2. A 10-g target turn in the azimuth plane of the radar.
3. A 10-g target turn "up" in the geometric plane perpendicular to the azimuth plane and the ground projection of the radar range vector.

The final, or third turn, starts when the target has rotated through 90.0 degrees of the 10-g turn in the azimuth plane. The maneuver was started at three initial ranges:

1. Distant : 11275.0 yards
2. Mid-range : 1000.0 yards
3. Close-in : 150.0 yards

Inputs to the filter of two noisy radar range measurements and an optional radial component of the predicted normal acceleration vector were generated at a sample rate of 10/second. A one sigma standard deviation Gaussian noise of 1.0 yard was added to the two range measurements. These inputs were input to the range filter.

Typical results of range error for one simulation of all maneuvers are listed in Table III. The addition of the predicted normal acceleration measurement to the radar range measurements reduced the peak and RMS errors for all maneuvers. The reduction in error was most dramatic for the close-in maneuver because of the range reversal for the particular azimuth and elevation geometries. At the mid-range and distant range trajectories there was no range reversal so the additional acceleration measurement was less helpful in correcting the range and range rate estimates In addition, the high radar sample rates and the averaging effect of using two radar measurements reduced the difference in error between the radar-only estimate and the combined estimate.

Table III.

Range Error (RMS Average/Peak - in Yards)

Maneuver	Combined Measurement (Normal Acceleration and Two Radars)	Two Radars Only
1. Distant	0.76/3.22	0.99/4.82
2. Mid-Range	0.89/4.25	1.06/5.53
3. Close-In	4.83/7.56	10.28/12.02

Fig. 19 shows the aircraft 10-g turn trajectory plotted in three dimensions for a 175 yards/second air speed. As compared to the 100 yards/second air speed trajectory shown in Fig. 20, the distance traveled down range is much greater, and the turning radius was much larger. However, within the 5 second interval of flight, the aircraft in Fig. 19 did not climb as high as that in Fig. 20. This is because the third maneuver (turn up) did not start until approximately 3 seconds versus 2 seconds for the 100 yards/second trajectory.

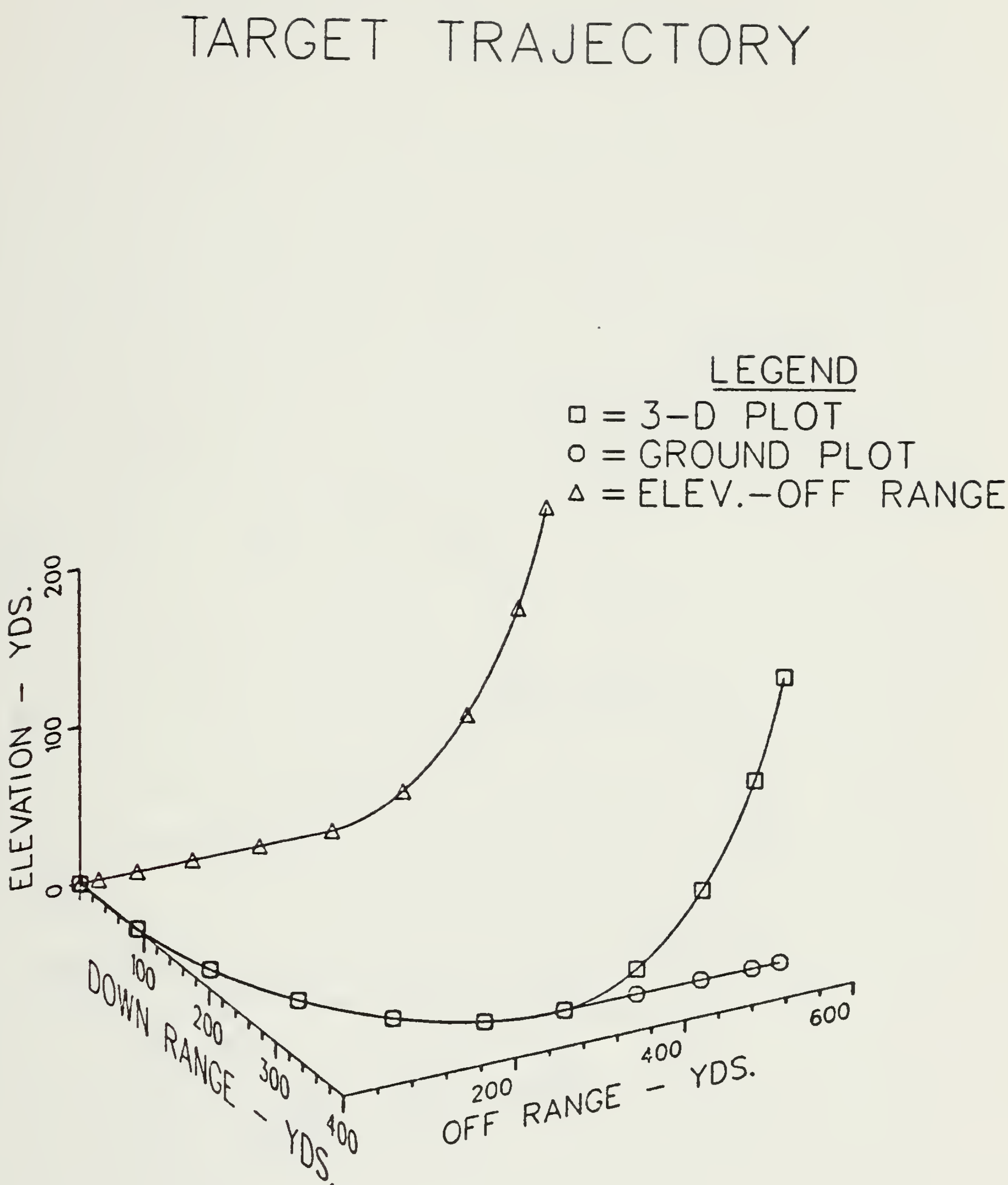

Figure 19

Three Dimensional Aircraft Trajectory,
Velocity = 175 yards/second

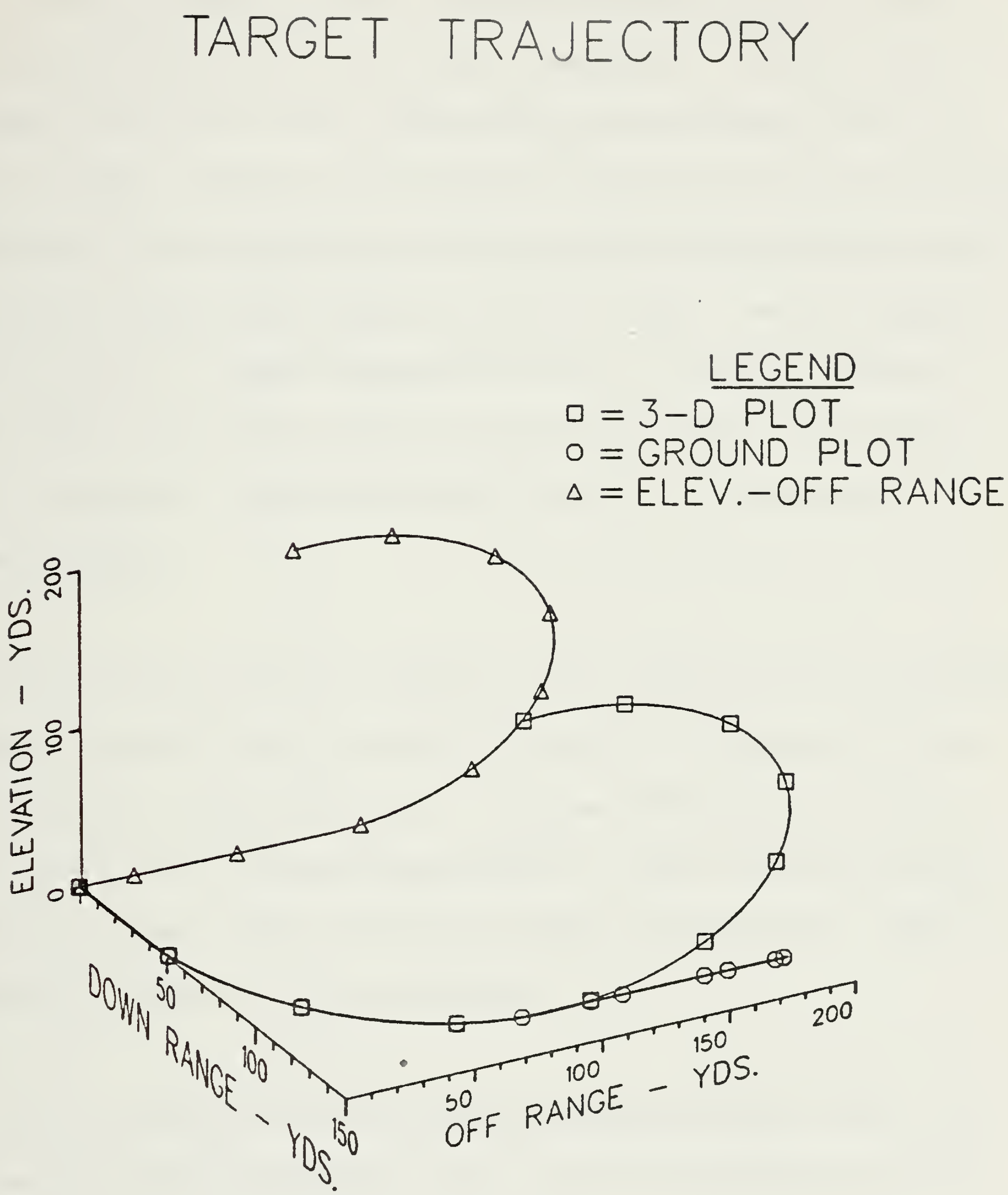

Figure 20

Three Dimensional Trajectory,
Velocity = 100 yards/second

In Fig. 21 the radial error and the measurement noise in yards is plotted versus time for the close-in maneuver. The radial error is equal to the true radial distance minus the estimated radial distance in yards. Gaussian noise with zero mean and a standard deviation proportional to the square root of the estimated radial range variance was the measurement noise added to the true range. For this example the noise has a 1 sigma standard deviation of 2.0 yards. The very large peak error at 0.8 seconds is due to the aircraft flying very nearly "overhead". This then reverses the sign of the relative velocity vector.

The velocity reversal is shown in Fig. 22 where X(2) is the Kalman estimator velocity state, and VR2D is the first difference of range divided by the sampling interval. The slow time constant of X(2) is due to the lack of a velocity measurement. A more accurate velocity estimate is shown in Fig. 23 for the distant turn maneuver. There is no velocity reversal in this trajectory.

Fig. 24 shows the range acceleration state estimate, X(3), and the measured range acceleration, Y(3). The two accelerations track quite closely and describe two phenomenon:

1. An initialization.

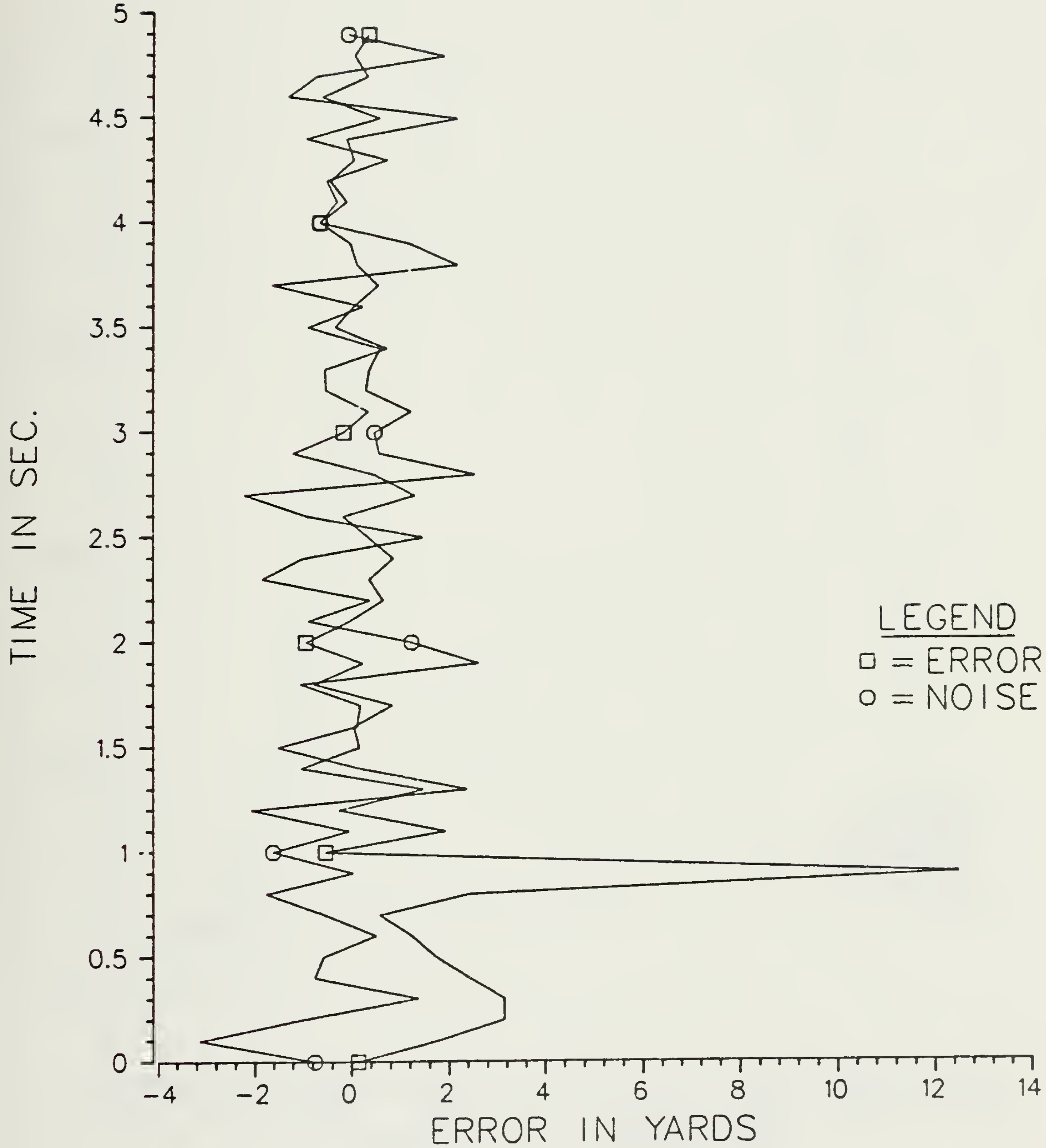

Figure 21

Range Error and Measurement Noise

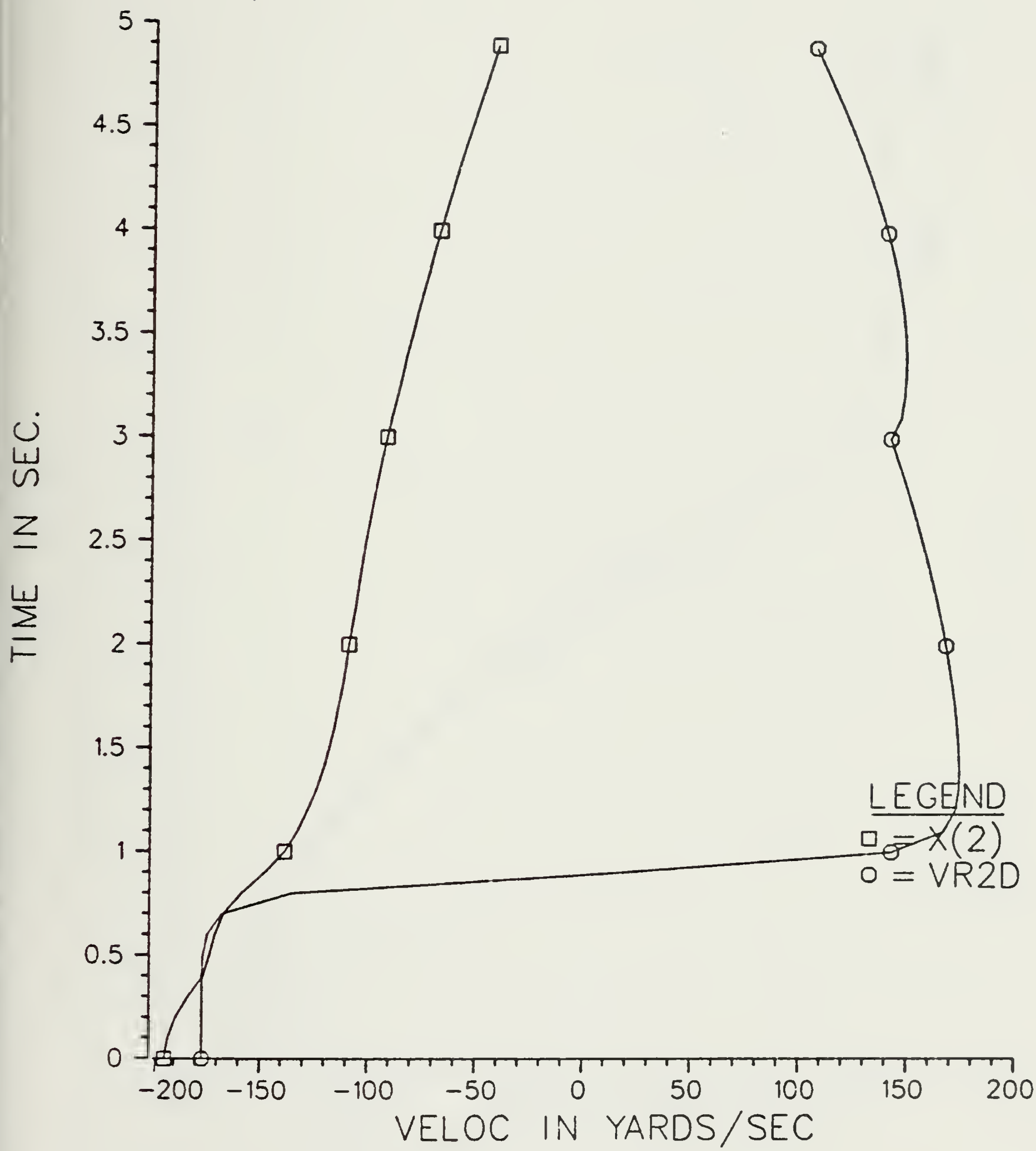

Figure 22

Estimated Range Velocity and Differential Range Divided by the Sampling Interval, Close-In Maneuver

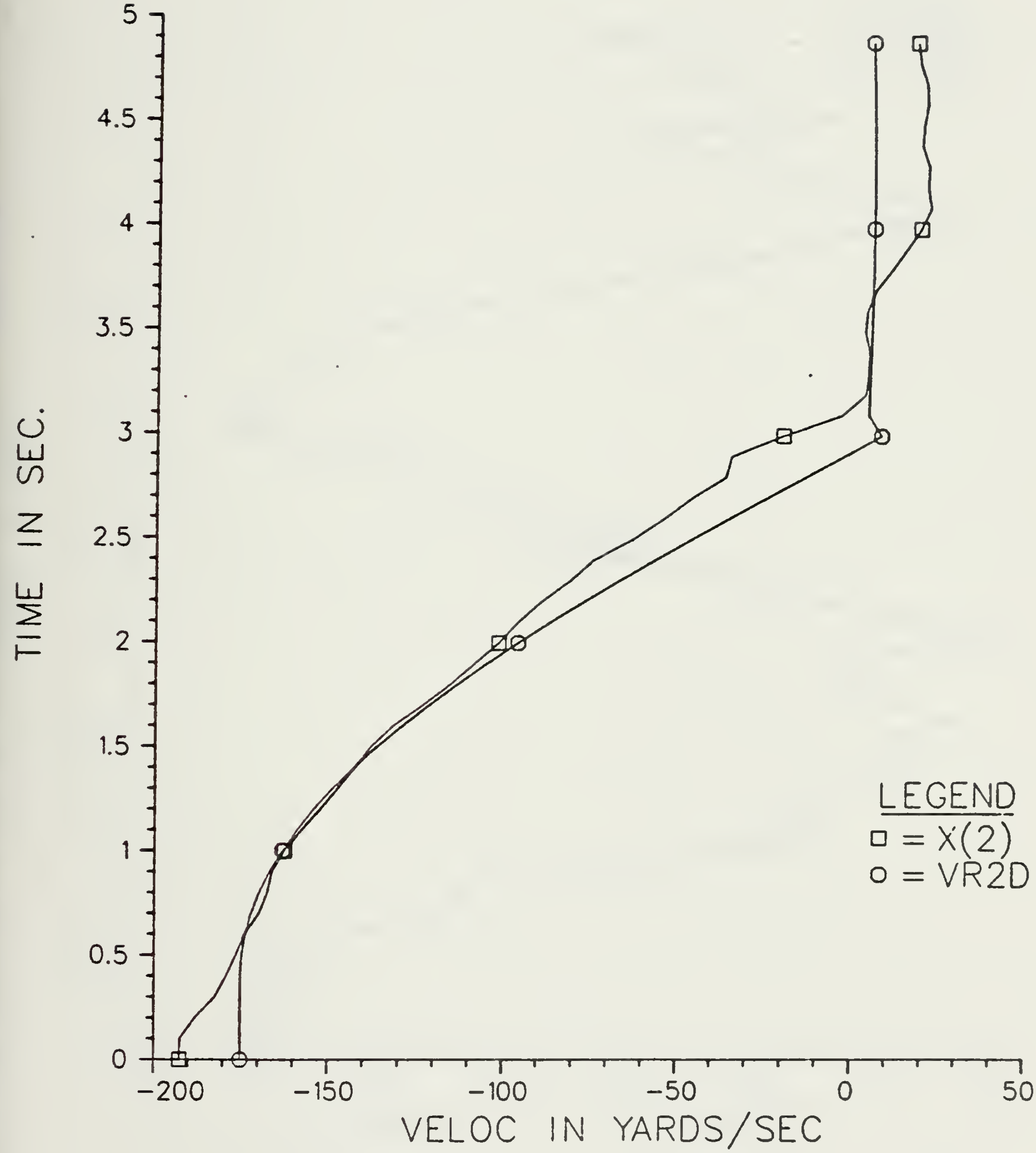

Figure 23

Estimated Range Velicity and Differential Range
Divided by the Sampling Interval, Distant Maneuver

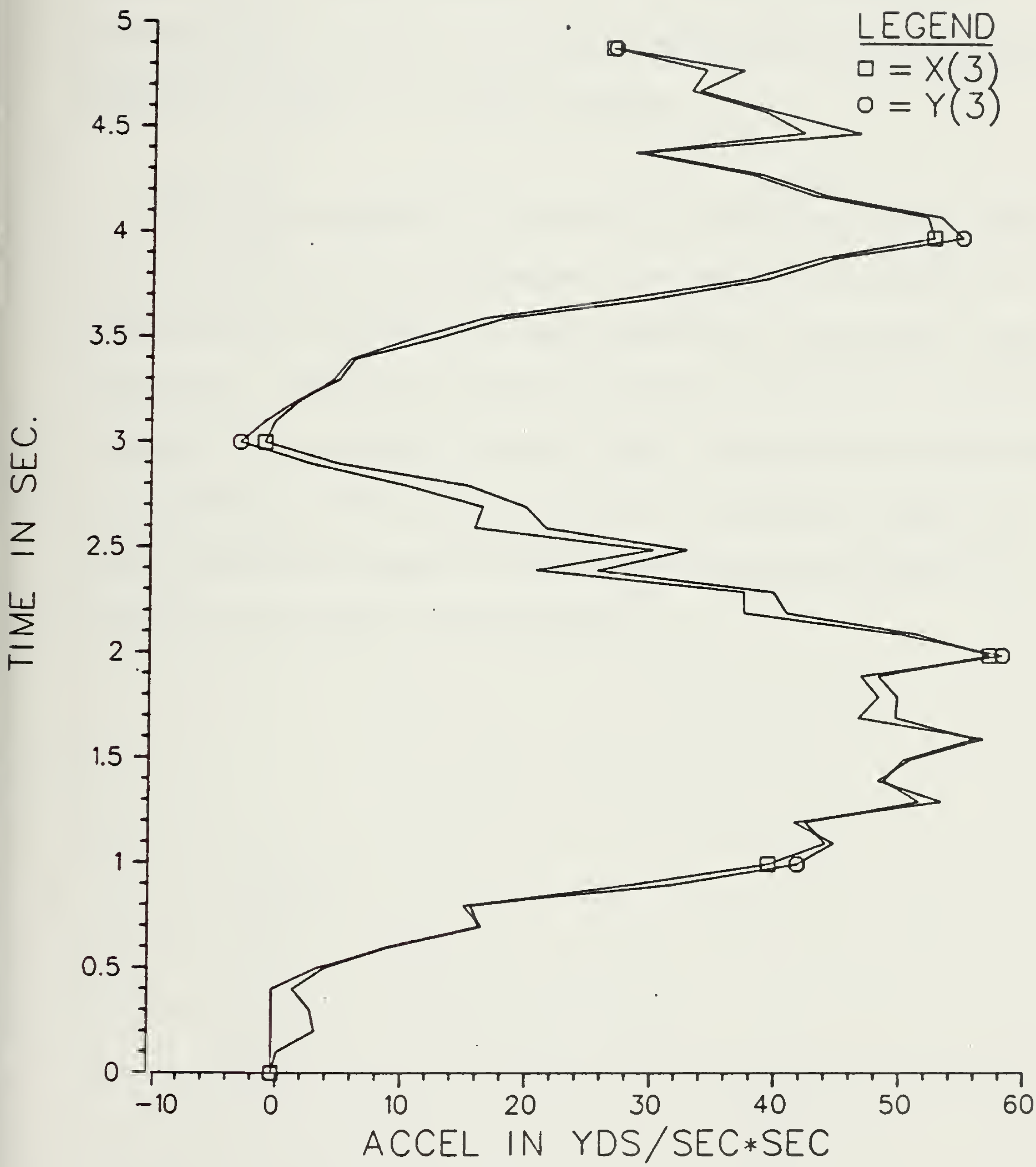

Figure 24

Range Acceleration Estimate and Range Acceleration Measurement

2. Cyclic acceleration matching the period of the third part of the maneuver.

The acceleration state estimate in Fig. 25 with no acceleration measurement does not show this periodic acceleration, indicating the uncertainty present in the state estimate.

This section applied the correlation attitude measurement technique to the radar tracking problem. The attitude is related to acceleration and therefore a useful measurement in predicting target acceleration. Recursive trajectory equations were used in a computer simulation that estimated aircraft range. The simulation demonstrated that the Kalman Filter was more accurate in tracking a target with varying trajectories when the predicted acceleration was used as in addition to the noisy range measurements.

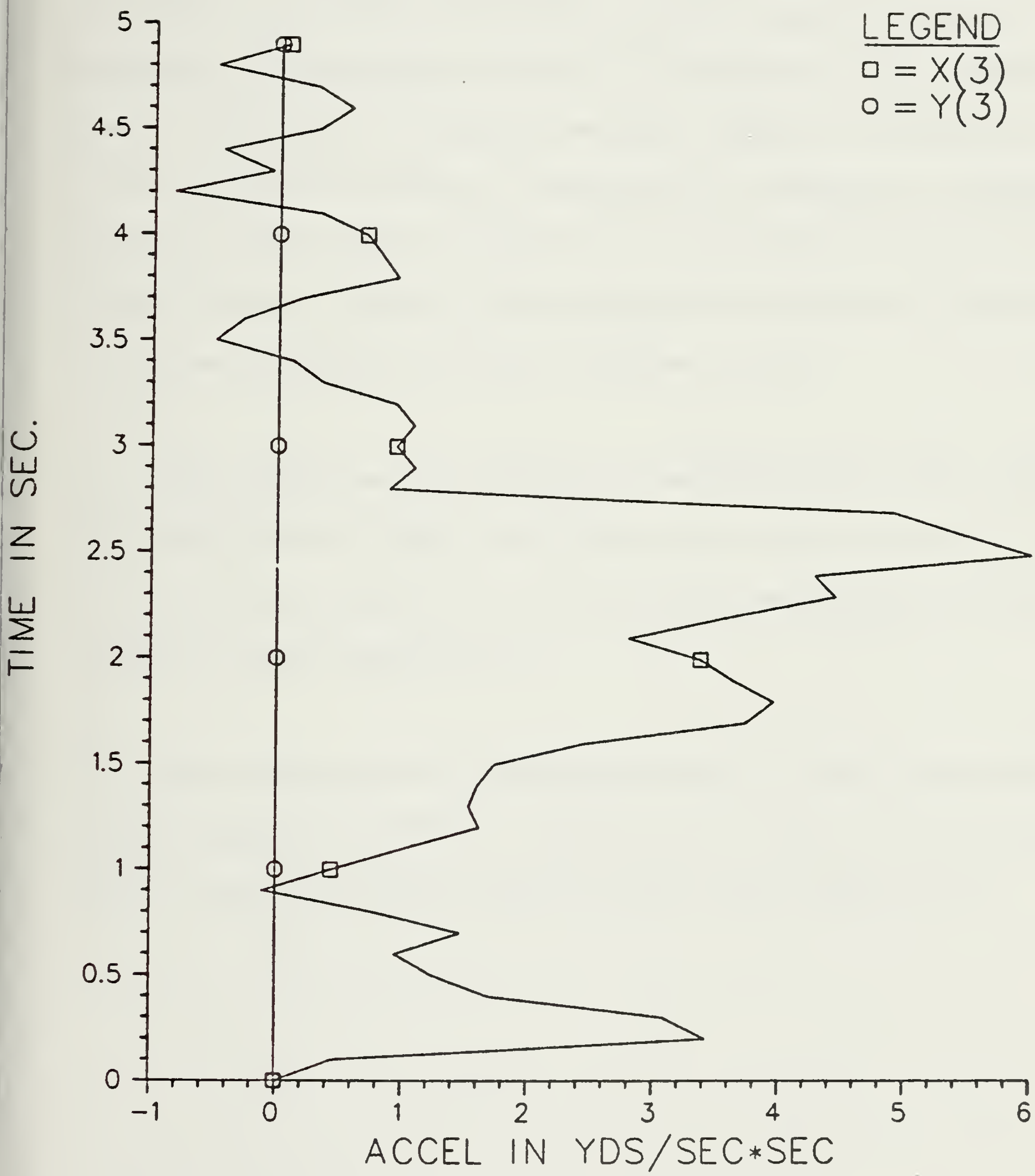

Figure 25

Range Acceleration Estimate Without Range Acceleration Measurement

VII. CONCLUSIONS

Manual optical image comparisons have been the primary method of measuring attitude of three-dimensional objects. Section V lists the general two-dimensional Fourier transform equations that are applied to an images edge data and a stored model of the solid object. The attitude of a three-dimensional object is estimated by using a three-axis quadratic interpolation to sequentially maximize the correlation output as a function of the attitude, position and size variables. Moreover, this measurement or estimate is equivalent to the maximum output of a bank of matched filters.

This concept was then applied to realistic Kalman Filter tracking simulations and shown to give improvement over conventional target state estimators that do not use the attitude measurement as a predictor of target acceleration.

Since the necessary attitude measurement is within the electro-optical tracker and computer state-of-the-art, the use of this estimator concept can improve present tracker system capabilities.

APPENDIX A

TWO-DIMENSIONAL DISCRETE CORRELATION

This appendix demonstrates that the two-dimensional discrete correlation transform (31), (32) is equivalent in both the time domain and the frequency domain.

$$c(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i,j)\, h(k+i,\, l+j) \qquad (31)$$

$$C(m,n) = X^*(m,n)\, H(m,n) \qquad (32)$$

where * = complex conjugate

Equations (31) and (32) are verified by performing the Fourier Transform and conjugation operations indicated.

$$c(k,l) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[\frac{1}{M\,N} \sum_{m=0} \sum_{n=0} X^*(m,n)\, e^{-j \frac{2\Pi}{M} i m}\, e^{-j \frac{2\Pi}{N} j n} \right]$$

$$\left[\frac{1}{PQ} \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} H(p,q)\, e^{j \frac{2\Pi}{M} p(k+i)}\, e^{j \frac{2\Pi}{N} q(l+j)} \right] \qquad \text{(A-1)}$$

$$c(k,l) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} X^*(m,n)\, H(p,q)\, e^{j \frac{2\Pi}{M} p k}\, e^{j \frac{2\Pi}{N} q l}$$

$$\left[\frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} e^{-j \frac{2\Pi}{M} i m}\, e^{j \frac{2\Pi}{N} j n}\, e^{j \frac{2\Pi}{M} p i}\, e^{j \frac{2\Pi}{N} q j} \right] \qquad \text{(A-2)}$$

The term in brackets in equation (A-2) is equal to:

$$\frac{1}{MN}(M\,N) = 1, \text{ if } m = p \text{ and } n = q$$

$$0 \quad , \text{ if } m \neq p \text{ and } n \neq q$$

due to orthogonality of i and j. Therefore:

$$c(k,l) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X^*(m,n)\, H(m,n)\, e^{j \frac{2\Pi}{M} m k}\, e^{j \frac{2\Pi}{N} n l}$$

$$= \mathcal{F}^{-1} \left[X^*(m,n)\, H(m,n) \right] \tag{A-3}$$

APPENDIX B

COMPUTER PROGRAMS

This Appendix is composed of the listings of the Fortran programs and subroutines for the attitude measurement and target tracking simulations. The programs were run on a Univac 1110 computer with a Fortran compiler and a FLECS translator (Fortran Language with Extended Control Structures). The graphical outputs were plotted using DISSPLA programs that generated COM-80 crt images.

```
      CCMR
      DIMENSION WH(16384),WHR(16384)
      DIMENSION S(128),INV(128),M(3)
      DIMENSION AZA(6),ELA(6),ROLLA(6),CMAX(20)
      DIMENSION Z(200)
      COMPLEX*8 A(16384)
      COMMON WHR,WH,A,CMAX,Z
      DO 111 J=1,20
        CMAX(J)=-1.0E10
  111 CONTINUE
      IPLOT=1
      IPRINT=1
    1 READ (5,10) AZ,EL,ROLL,LAST
   10 FORMAT()
      WRITE(6,101) AZ,EL,ROLL,LAST
  101 FORMAT(1X,3F9.2,I4)
      IF(LAST.EQ.1)  IPLOT=3
C
C      READ ARRAY SIZE IN POWERS OF 2
C     READ NUMBER OF ARRAYS TO BE CORRELATED (MAX OF 20)
C
      READ(5,5)KK,JJ
  5   FORMAT(2I5)
      READ(5,5) NA
      N1=2**KK
      N2=2**JJ
      N3=1
      KK2=KK+1
      JJ2=JJ+1
      N12=2*N1
      N22=2*N2
      NH=N1/2
      NJ=N2/2+.5
      NV=N1*N2
      NV4=N12*N22
      NC=(N1*N2)/24+1
      M(1)=KK2
      M(2)=JJ2
      M(3)=0
      PI=3.14159
      WRITE(6,15) N1,N2
  15  FORMAT('      INPUT ARRAY DIMENSIONS = (',I3,'X',I3,')',/)
C
C     READ IN THE DATA ARRAY
C
      WRITE(6,8)
      DO 11 K=1,NC
      K1=(K-1)*24+1
      K24=K*24
      READ(5,7)(WH(I),I=K1,K24)
      WRITE(6,9)K1,(WH(I),I=K1,K24)
  11  CONTINUE
   7  FORMAT(24F3.0)
```

```
  8   FORMAT('    I   ',24('WH(I)'),//)
  9   FORMAT(1X,I5,24(1X,F4.0))
C
C     DOUBLE SIZE OF DATA ARRAY
C     WH = INPUT VECTOR, WHR = OUTPUT VECTOR, NV = INPUT DIMENSION
C
      CALL VDOUBL(WH,WHR,NV,NV4)
C
C     PLOT DATA ARRAY
C
C
C     TRANSFER THE WH(K) DATA TO THE COMPLEX A MATRIX
C
      DO 3 I2=1,N22
      DO 3 I1=1,N12
      IWH= (I2-1)*N1+I1
      A(IWH)=WHR(IWH)
  3   CONTINUE
C
C     COMPUTE THE DISCRETE FOURIER TRANSFORM OF THE WH(K) DATA
C
      CALL HARM(A,M,INV,S,1,IPERR)
C
C     READ IN THE SECOND DATA ARRAY
C
C
C  ROTATE MODEL IN AZMUTH
C
      NL=0
      NA1=1
      DELAL =5.
      AZA(1)=AZ
      AZA(2)=AZ-DELAL
      AZA(3)=AZ+   DELAL
 290  CONTINUE
      DO 300 J=NA1,NA
C
C  GENERATE IMAGE BOUNDARY FROM 3-DIMENTIONAL MODEL
C  WHR = CARD = OUTPUT VECTOR (INCYM X INCYM)
C
      CALL SURFAC(AZA(J),EL,ROLL,YMAX,ZMAX,YMIN,ZMIN,J,IPLOT,IPRINT)
C
C  CALCULATE CORRELATION BETWEEN INPUT IMAGE AND GENERATED IMAGE BOUNDA
C
2037  FORMAT(/,5X,'J =',I2,3X,'CMAX =',F17.6,/)
      CALL CALCOR(KK2,JJ2,NV ,AZA(J),EL,ROLL,J,N12,N22)
 300  CONTINUE
      NL=NL+1
      IF(NL.GT.6) GO TO 3202
      WRITE(6,310)
 310  FORMAT(/,5X,'J',3X,'AZA(J)',7X,'CMAX(J)',/)
      DO 311 J=1,3
      WRITE(6,312) J,AZA(J),CMAX(J)
 311  CONTINUE
```

```
  312 FORMAT(4X,I2,3X,F6.2,3X,F11.6)
C
C   TEST FOR MAXIMUM CORRELATION. IF CMAX(1) < CMAX(2) OR CMAX(3)
C   CORRELATE AT ADDITIONAL POINTS
C
      IF(CMAX(1).LT.CMAX(2)) GO TO 3141
      IF(CMAX(1).LT.CMAX(3)) GO TO 3142
      GO TO 3144
 3141 IF(CMAX(2).LT.CMAX(3)) GO TO 3142
      AZA(3)=AZA(1)
      AZA(1)=AZA(2)
      AZA(2)=AZA(2)-DELAL
      CMAX(3)=CMAX(1)
      CMAX(1)=CMAX(2)
      CMAX(2)=-1.E10
      NA1=2
      NA=2
      GO TO 290
 3142 CONTINUE
      AZA(2)=AZA(1)
      AZA(1)=AZA(3)
      AZA(3)=AZA(3)+DELAL
      CMAX(2)=CMAX(1)
      CMAX(1)=CMAX(3)
      CMAX(3)=-1.E10
      NA1=3
      NA=3
      GO TO 290
 3144 CONTINUE
C
C     CALCULATE MOST PROBABLE AZMUTH FOR INPUT FRAME BY QUDRATIC INTERP
C
      AZST=(CMAX(2)-CMAX(3))*DELAL /(2.*CMAX(3)+2.*CMAX(2)-4.*CMAX(1))
      IAZST=IFIX(AZST+.5)+IFIX(AZA(1))
 2222 FORMAT(2A6)
      PRINT 320,AZST,IAZST
  320 FORMAT(/,5X,'AZST =',F12.5,5X,'IAZST =',I4,/)
C
C   REDUCE INTERVAL OF INTERPOLATION IF ABS. VALUE OF AZST > 1.
C
      IF(ABS(AZST).LT.1.0) GO TO 3202
      DELAL=2.0
      NA1=2
      NA=3
      AZA(2)=AZA(1)-DELAL
      AZA(3)=AZA(1)+DELAL
      CMAX(2)=-1.E10
      CMAX(3)=-1.E10
      GO TO 290
 3202 CONTINUE
      AZ=FLOAT(IAZST)
C
C   ROTATE MODEL IN ELEVATION
C
```

```
      DELEPS=5.
      NL=0
      CMAX(7)=CMAX(1)
      NA8=8
      NA9=9
      ELA(1)=EL
      ELA(2)=EL-DELEPS
      ELA(3)=EL+DELEPS
  390 CONTINUE
      DO 400 J=NA8,NA9
      JM=J-6
C
C  GENERATE IMAGE BOUNDARY FROM 3-DIMENTIONAL MODEL
C  WHR = CARD = OUTPUT VECTOR (INCYM X INCYM)
C
      CALL SURFAC(AZ,ELA(JM),ROLL,YMAX,ZMAX,YMIN,ZMIN,J,IPLOT,IPRINT)
C
C  CALCULATE CORRELATION BETWEEN INPUT IMAGE AND GENERATED IMAGE BOUNDA
C
      CALL CALCOR(KK2,JJ2,NV ,AZ,ELA(JM),ROLL,J,N12,N22)
  400 CONTINUE
      NL=NL+1
      IF(NL.GT.6) GO TO 4202
      WRITE(6,410)
  410 FORMAT(/,5X,'J',3X,'ELA(JM)',7X,'CMAX(J)',/)
      DO 411 J=7,9
      JM=J-6
      WRITE(6,312) J,ELA(JM),CMAX(J)
  411 CONTINUE
C
C  TEST FOR MAXIMUM CORRELATION. IF CMAX(7) < CMAX(8) OR CMAX(9)
C  CORRELATE AT ADDITIONAL POINTS
C
      IF(CMAX(7).LT.CMAX(8)) GO TO 4141
      IF(CMAX(7).LT.CMAX(9)) GO TO 4142
      GO TO 4144
 4141 IF(CMAX(8).LT.CMAX(9)) GO TO 4142
      ELA(3)=ELA(1)
      ELA(1)=ELA(2)
      ELA(2)=ELA(2)-DELEPS
      CMAX(9)=CMAX(7)
      CMAX(7)=CMAX(8)
      CMAX(8)=-1.E10
      NA8=8
      NA9=8
      GO TO 390
 4142 CONTINUE
      ELA(2)=ELA(1)
      ELA(1)=ELA(3)
      ELA(3)=ELA(3)+DELEPS
      CMAX(8)=CMAX(7)
      CMAX(7)=CMAX(9)
      CMAX(9)=-1.E10
      NA9=9
```

```
      NA8=9
      GO TO 390
 4144 CONTINUE
C
C     CALCULATE MOST PROBABLE ELEVATION FOR INPUT FRAME BY QUDRATIC INT
C
      ELST=(CMAX(8)-CMAX(9))*DELEPS/(2.*CMAX(9)+2.*CMAX(8)-4.*CMAX(1))
      IELST=IFIX(ELST+.5)+IFIX(ELA(1))
      PRINT 321,ELST,IELST
  321 FORMAT(/,5X,'ELST =',F12.5,5X,'IELST =',I4,/)
C
C  REDUCE INTERVAL OF INTERPOLATION IF ABS. VALUE OF ELST > 1.
C
      IF(ABS(ELST).LT.1.0) GO TO 4202
      DELEPS=2.
      NA8=8
      NA9=9
      ELA(2)=ELA(1)-DELEPS
      ELA(3)=ELA(1)+DELEPS
      CMAX( 8)=-1.E10
      CMAX( 9)=-1.E10
      GO TO 390
 4202 CONTINUE
      EL=FLOAT(IELST)
C
C  ROTATE MODEL IN ROLL
C
      DELRHO=5.
      NL=0
      CMAX(13)=CMAX(7)
      ROLLA(1)=ROLL
      ROLLA(2)=ROLL-DELRHO
      ROLLA(3)=ROLL+DELRHO
      NA14=14
      NA15=15
  490 CONTINUE
      DO 500 J=NA14,NA15
      JM=J-12
C
C  GENERATE IMAGE BOUNDARY FROM 3-DIMENTIONAL MODEL
C  WHR = CARD = OUTPUT VECTOR (INCYM X INCYM)
C
      CALL SURFAC(AZ,EL,ROLLA(JM),YMAX,ZMAX,YMIN,ZMIN,J,IPLOT,IPRINT)
C
C  CALCULATE CORRELATION BETWEEN INPUT IMAGE AND GENERATED IMAGE BOUNDA
C
      CALL CALCOR(KK2,JJ2,NV ,AZ,EL,ROLLA(JM),J,N12,N22)
  500 CONTINUE
      NL=NL+1
      IF(NL.GT.6) GO TO 5202
      WRITE(6,510)
  510 FORMAT(/,5X,'J',3X,'ROLLA(JM)',7X,'CMAX(J)',/)
      DO 511 J=13,15
      JM=J-12
```

```
      WRITE(6,312) J,ROLLA(JM),CMAX(J)
  511 CONTINUE
C
C  TEST FOR MAXIMUM CORRELATION. IF CMAX(13) < CMAX(14) OR CMAX(15)
C  CORRELATE AT ADDITIONAL POINTS
C
      IF(CMAX(13).LT.CMAX(14)) GO TO 5141
      IF(CMAX(13).LT.CMAX(15)) GO TO 5142
      GO TO 5144
 5141 IF(CMAX(14).LT.CMAX(15)) GO TO 5142
      ROLLA(3)=ROLLA(1)
      ROLLA(1)=ROLLA(2)
      ROLLA(2)=ROLLA(2)-DELRHO
      CMAX(15)=CMAX(13)
      CMAX(13)=CMAX(14)
      CMAX(14)=-1.E10
      NA14=14
      NA15=14
      GO TO 490
 5142 CONTINUE
      ROLLA(2)=ROLLA(1)
      ROLLA(1)=ROLLA(3)
      ROLLA(3)=ROLLA(3)+DELRHO
      CMAX(14)=CMAX(13)
      CMAX(13)=CMAX(15)
      CMAX(15)=-1.E10
      NA14=15
      NA15=15
      GO TO 490
 5144 CONTINUE
C
C     CALCULATE MOST PROBABLE   ROLL FOR INPUT FRAME BY QUDRATIC INTERP
C
      ROLLST=(CMAX(14)-CMAX(15))*DELRHO/(2.*CMAX(15)+2.*CMAX(14)
     *-4.*CMAX(1))
       IROLST=IFIX(ROLLST+.5)+IFIX(ROLLA( 1))
      PRINT 323,ROLLST, IROLST
  323 FORMAT(/,5X,'ROLLST =',F12.5,5X,' IROLST =',I4,/)
C
C  REDUCE INTERVAL OF INTERPOLATION IF ABS. VALUE OF ROLLST > 1.
C
      IF(ABS(ROLLST).LT.1.0) GO TO 5202
      DELRHO=2.
      NA14=14
      NA15=15
      ROLLA(2)=ROLLA(1)-DELRHO
      ROLLA(3)=ROLLA(1)+DELRHO
      CMAX(14)=-1.E10
      CMAX(15)=-1.E10
      GO TO 490
 5202 CONTINUE
      IPLOT=2
      IF(LAST.EQ.0) GO TO 1
      STOP
```

SURFAC

```
C SURFAC ROTATES A STORED 3-DIMENTIONAL MODEL IN 3 AXIS  AZMUTH.ELEVAT-
C ION THE PROJECTED IMAGE OF THE MODEL ON A PLANE SURFACE IS GUANTIZED
C ONTO A (32X32) GRID.  OUTPUT ARRAY = 'CARD'

      SUBROUTINE SURFAC(AZ.EL,ROLL.YMAX1.ZMAX1.YMIN1.ZMIN1.JA.IPLOT.
     . IPRINT)
      PARAMETER NCS=71.K1=2*NCS
      PARAMETER INCYM  =58
      PARAMETER IZE=C.IE=1
      PARAMETER NI=5.NO=6.NOP=7
      DIMENSION AKMAX(INCYM).P(3.4.NCS).T(3.5.NCS).XNORM(NCS).YK(2.NCS).
     AYNORM(NCS).ZNORM(NCS),AKMIN(INCYM),                   BKMAX(INCYM  ).
     B BKMIN(INCYM   )
      DIMENSION CARD(16384)
      DIMENSION YL(10).ZL(10)
      COMMON CARD
      COMMON/FACDAT/ P
      ZERO=0.0
      ONE=1.
      LCTR=0
      DRC=ACOS(-1.)/180.
C
C GENERAL PREPERATION OF DATA
C
      AZR=AZ*DRC
      ELR=EL*DRC
      ROLLR=ROLL*DRC
      SAZ=SIN(AZR)
      CAZ=COS(AZR)
      SEL=SIN(ELR)
      CEL=COS(ELR)
      SRO=SIN(ROLLR)
      CRO=COS(ROLLR)
      YMAX=-10.E10
      YMIN=10.E10
      ZMAX=-10.E10
      ZMIN=10.E10
 1222 FORMAT(1X.I5.12F10.2)
C
C ROTATE TARGET
C
      DO 10 K=1.NCS
      DO 10 J=1.4
      T(1.J.K)=P(1.J.K)*CEL*CAZ-P(2.J.K)*CEL*SAZ+P(3.J.K)*SEL
      T(2,J,K)=P(1.J.K)*(CRO*SAZ+SRO*SEL*CAZ)+P(2,J,K)*(CRO*CAZ-SRO*SEL*
     1SAZ)-P(3.J.K)*SRO*CEL
      T(3.J.K)=P(1.J.K)*(SRO*SAZ-CRO*SEL*CAZ)+P(2.J.K)*(CRO*SEL*SAZ
     1   +SRO*CAZ)+P(3.J.K)*CRO*CEL
      IF(T(2.J,K).GT.YMAX) YMAX=T(2,J,K)
```

```
  IF(T(2,J,K).LT.YMIN) YMIN=T(2,J,K)
  IF(T(3,J,K).GT.ZMAX) ZMAX=T(3,J,K)
  IF(T(3,J,K).LT.ZMIN) ZMIN=T(3,J,K)
) CONTINUE

AVE FIRST YM-- AND ZM-- VALUES
  LCTR=LCTR+1
  IF(JA.NE.1) GO TO 12
  YMAX1=YMAX
  YMIN1=YMIN
  ZMAX1=ZMAX
  ZMIN1=ZMIN
2 CONTINUE

MPUTE SURFACE NORMALS :

   DO 15 K=1,NCS
   XNORM(K)=(T(2,1,K)-T(2,2,K))*(T(3,1,K)-T(3,4,K))-(T(2,1,K)-
  1      T(2,4,K))*(T(3,1,K)-T(3,2,K))
   YNORM(K)=(T(3,1,K)-T(3,2,K))*(T(1,1,K)-T(1,4,K))-(T(3,1,K)    -
  1      T(3,4,K))*(T(1,1,K)-T(1,2,K))
   ZNORM(K)=(T(1,1,K)-T(1,2,K))*(T(2,1,K)-T(2,4,K))-(T(1,1,K)   -
  1      T(1,4,K))*(T(2,1,K)-T(2,2,K))
15 CONTINUE

OMPUTE IYMAX AND IZMAX

   LCTR=LCTR+1
   RZMAX=ABS(ZMAX-ZMIN)
   RYMAX=ABS(YMAX-YMIN)
   IZMAX=RZMAX
   IYMAX=RYMAX

PREPARE PLOT PAGE

   IF(IPLOT.EQ.0) GO TO 19
   CALL TMPPLT(9)
   AXLGTH= AMAX1((ZMAX -ZMIN ),(YMAX -YMIN ))
   AXLGTH = AINT(AXLGTH/2.) + 10
   XCNTR=AINT((YMAX+YMIN)/2)
   YCNTR=AINT((ZMAX+ZMIN)/2)
   XORG=XCNTR-AXLGTH
   YORG=YCNTR-AXLGTH
   XEND=XCNTR+AXLGTH
   YEND=YCNTR+AXLGTH
   CALL BGNPL(-1)
   CALL NOBRDR
   CALL HEIGHT(.1)
   ENCODE(60,210,LABEL)AZ,EL,ROLL
   CALL TITLE(LABEL,100,' ',1,' ',1,6.,6.)
   CALL GRAF(XORG,5,XEND,YORG,5,YEND)
   CALL MARKER(0)
   CALL SCLPIC(.06)
210 FORMAT('           AZ=',F6.1,3X,'EL=',F6.1,3X,'ROLL=',F6.1,'$')
```

```
 10 CONTINUE
C
C START INDEX FOR Z-RASTER
C
    RNCYM=FLOAT(INCYM)
    DLY=RYMAX/RNCYM
    DLZA=RZMAX/RNCYM
 51 FORMAT('    PLT1.  PLT2.'26X.'YMIN.'.10X.'ZMIN'.14X.'DLY.'.10X.
   B 'DLZA'. 8X.'IZMAX.'. 1X.'IYMAX'./)
 52 FORMAT(6F16.5,2I5./)
    KMAX=0
    KMIN=0
    SEED=1.
    ISEED=1
    WRITE(6.1222) LCTR
    DO 100 IZ=1.INCYM
    ZBAR=IZ*DLZA+ZMIN
C
C FIND Y INTERSECTIONS
C
    LVE=0
    AKMAX(IZ)=-10.E10
    AKMIN(IZ)=10.E10
251 FORMAT('  Y. ZBAR. AKMIN(IZ). AKMAX(IZ).DYMIN.ZYMIN(INCY).DYMAX.
   .ZYTEMP.MAX(INCY).'.20X.'IZ.I.   IFLG.IAKFLG(INCY).INCY'./)
    DO 40 I=1,NCS
    YK(1.I)=YMIN-100.
    IF(XNORM(I).LE.0.) GO TO 40
    NBR=0
    DO 25 K=1.4
    KE=K+1
    IF(K.EQ.4) KE=1
    IF(ABS(T(3,K,I)-T(3,KE,I)).LE.1.E-6) GO TO 25
    Y =(T(2,KE.I)-T(2.K.I))*(ZBAR-T(3.K,I))/(T(3.KE.I)-T(3,K,I))+
   1      T(2.K.I)
    X =(T(1.KE,I)-T(1.K.I))*(ZBAR-T(3.K,I))/(T(3.KE,I)-T(3.K.I))+
   A  T(1,K,I)
    INCY=(Y-YMIN)/DLY
C
C IS INTERSECTION WITHIN BOUNDS OF THE SURFACE
C
    IF(Y.GT.T(2,K,I).AND.Y.GT.T(2.KE.I)) GO TO 25
    IF(Y.LT.T(2.K.I).AND.Y.LT.T(2.KE.I)) GO TO 25
    IF(ZBAR.LT.T(3.K.I).AND.ZBAR.LT.T(3.KE.I)) GO TO 25
    IF(ZBAR.GT.T(3,K,I).AND.ZBAR.GT.T(3,KE.I)) GO TO 25
    IF(Y.LE.AKMAX(IZ)) GO TO 195
    AKMAX(IZ)=Y
195 IF(Y.GE,AKMIN(IZ)) GO TO 196
    AKMIN(IZ)=Y
196 CONTINUE
 20 CONTINUE
 25 CONTINUE
 40 CONTINUE
C
```

```
 ~LU HANDOM NCISE TO AKM--(IZ)

PLOT POINTS YL(I)

    IF(IPLOT.EQ.0) GO TO 65
    ZL(1)=ZBAR
    ZL(2)=ZBAR
    WHEN(AKMAX(IZ).LT.PLT1.OR.AKMAX(IZ).GT.PLT2)
    .   UNLESS(AKMIN(IZ).LT.PLT1.OR.AKMIN(IZ).GT.PLT2)
    .   .   YL(1)=AKMIN(IZ)
    .   .   KMIN=KMIN+1
    .   .   CALL CURVE(YL.ZL. 1.-1)
    .   ...FIN
    ...FIN
    ELSE
    .   WHEN(AKMIN(IZ).LT.PLT1.OR.AKMIN(IZ).GT.PLT2)
    .   .   YL(1)=AKMAX(IZ)
    .   .   KMAX=KMAX+1
    .   .   CALL CURVE(YL.ZL. 1.-1)
    .   ...FIN
    .   ELSE
    .   .   YL(1)=AKMAX(IZ)
    .   .   YL(2)=AKMIN(IZ)
    .   .   KMAX=KMAX+1
    .   .   KMIN=KMIN+1
    .   .   CALL CURVE(YL.ZL. 2.-1)
    .   ...FIN
    ...FIN
 60 CONTINUE
 65 CONTINUE
100 CONTINUE
    LCTR=LCTR+1
    JMIN=0
    JMAX=0

START INDEX FOR Y-RASTER

    LMAX=0
    LMIN=0
    WRITE(6,1222)  LCTR
    DO 500 I =1,INCYM
    YBAR=I *DLY +YMIN

FIND Z INTERSECTIONS

    BKMAX(I )=-10.E10
    BKMIN(I )=10.E10
    DO 400 J=1,NCS
    IF(XNORM(J).LE.0.) GO TO 400
    DO 300 K=1.4
    KE=K+1
    IF(K.EQ.4) KE=1
    IF(ABS(T(2.K.J)-T(2.KE.J)).LE.1.E-6) GO TO 300
```

```
C     AKM--(I)=HORIZONTAL LIMITS FOR SEQUENTIAL Z SCAN
C     BKM--(J)=VERTICAL LIMITS FOR SEQUENTIAL Y SCAN
C
C     CENTER IMAGE  WITH RESPECT TO  FIRST IMAGE AND USE GRID INTERVALS OF
C     FIRST IM SCAN IMAGE BOTTOM TO TOP (I) AND LEFT TO RIGHT (J) AT
C     DISCRETE INTERVALS.
C     CONVERT MAX. AND MIN. EDGE COORDINATES TO GRID INTEGERS.
C
      DEYMIN=YMIN-YMIN1
      DEYMAX=YMAX-YMAX1
      DELY=( DEYMIN+DEYMAX)/2.
      YMIN=YMIN1+DELY
      DLY1=ABS(YMAX1-YMIN1)/RNCYM
      DEZMIN=ZMIN-ZMIN1
      DEZMAX=ZMAX-ZMAX1
      DELZ=( DEZMIN+DEZMAX)/2.
      ZMIN=ZMIN1+DELZ
      DLZA1=ABS(ZMAX1-ZMIN1)/RNCYM
      INCTR=0
      DO 1050 I=1,INCYM
      FX=(AKMAX(I)-YMIN)/DLY1
      IX=IFIX(FX)
      GX=(AKMIN(I)-YMIN)/DLY1
      IG=IFIX(GX)
C     24 CHARACTER FLAG
      IRF=0
C     CONTRAST PIXEL FLAG
      IEF=0
      DO 1060 J=1,INCYM
      FY=(BKMAX(J)-ZMIN)/DLZA1
      JF=IFIX(FY)
      GY=(BKMIN(J)-ZMIN)/DLZA1
      JG=IFIX(GY)
      IR   =(I-1)*INCYM+J
      IF(IR.NE.0) GO TO 1030
      IR=24
      IRF=1
 1030 CONTINUE
      IF(IX.NE.J) GO TO 1033
        CARD(IR)=ONE
      IEF=1
      GO TO 1044
 1033 CONTINUE
        CARD(IR)=ZERO
      IF(IG.NE.J) GO TO 1036
        CARD(IR)=ONE
      IEF=1
      INCTR=INCTR+1
      GO TO 1044
 1036 CONTINUE
        CARD(IR)=ZERO
      IF(JF.NE.I) GO TO 1040
        CARD(IR)=ONE
      IEF=1
```

```
      ZF=(T(3.KE.J)-T(3.K.J))*(YBAR-T(2.K.J))/(T(2.KE.J)-T(2.K.J))+
     A  T(3.K.J)
C
C IS INTERSECTION WITHIN BOUNDS OF THE SURFACE
C
      IF(ZF.LT.T(3.K,J).AND.ZF.LT.T(3.KE.J)) GO TO 300
      IF(ZF.GT.T(3.K,J).AND.ZF.GT.T(3,KE,J)) GO TO 300
      IF(YBAR.GT.T(2.K.J).AND.YBAR.GT.T(2.KE.J)) GO TO 300
      IF(YBAR.LT.T(2.K.J).AND.YBAR.LT.T(2.KE.J)) GO TO 300
      IF(ZF.LE.BKMAX(I )) GO TO 250
      BKMAX(I )=ZF
      LMAX=LMAX+1
250   CONTINUE
      IF(ZF.GE.BKMIN(I )) GO TO 255
      BKMIN(I )=ZF
      LMIN=LMIN+1
255   CONTINUE
300   CONTINUE
400   CONTINUE
C
C     ADD RANDOM NOISE TO BKM--(I2)
C
C
C PLOT POINTS
C
      IF(IPLOT.EQ.0) GO TO 485
      YL(2)=YBAR
      YL(1)=YBAR
      WHEN(BKMAX(I ).LT.PLT3.OR.BKMAX(I ).GT.PLT4)
      .  UNLESS(BKMIN(I ).LT.PLT3.OR.BKMIN(I ).GT.PLT4)
      .  .  ZL(1)=BKMIN(I )
      .  .  JMIN=JMIN+1
      .  .  CALL CURVE(YL.ZL. 1,-1)
      .  ...FIN
      ...FIN
      ELSE
      .  WHEN  (BKMIN(I ).LT.PLT3.OR.BKMIN(I ).GT.PLT4)
      .  .  ZL(1)=BKMAX(I )
      .  .  JMAX=JMAX+1
      .  .  CALL CURVE(YL.ZL. 1.-1)
      .  ...FIN
      .  ELSE
      .  .  ZL(1)=BKMAX(I )
      .  .  ZL(2)=BKMIN(I )
      .  .  JMAX=JMAX+1
      .  .  JMIN=JMIN+1
      .  .  CALL CURVE(YL.ZL. 2.-1)
      .  ...FIN
      ...FIN
480   CONTINUE
485   CONTINUE
500   CONTINUE
      LCTR=LCTR+1
```

```
      GO TO 1044
1040  CONTINUE
        CARD(IR)=ZERO
      IF(JG.NE.I) GO TO 1043
        CARD(IR)=ONE
      IEF=1
      GO TO 1044
1043  CONTINUE
        CARD(IR)=ZERO
1044  CONTINUE
1065  CONTINUE
1050  CONTINUE
      LCTR=LCTR+1
      IF(IPRINT.EQ.0) GO TO 110
      PRINT 200,AZ,EL,ROLL
      CALL ENDPL(0)
      LCTR=LCTR+1
 110  CONTINUE
      LCTR=LCTR+1
      RETURN
 200  FORMAT(1H1,                        5X,'AZIMUTH ANGLE=',F8.2,/,5X,
    A 'ELEVATION ANGLE=',F8.2,/,5X,'ROLL ANGLE=',F8.2,//)
      END
```

```
      SUBROUTINE VDOUBL(ARIN,AROT,INDM,NOD)
C
C  VDOUBL CONVERTS AN INPUT (NXN) ARRAY INTO A (2NX2N) ARRAY BY ADDING
C  BACKGROUND DATA.
C     ARIN=INPUT ARRAY, AROT=OUTPUT ARRAY, INDM=INPUT DIMENTION
C
      DIMENSION ARIN(INDM), AROT(NOD)
      NODD=2*INDM
      NOD=4*INDM
      FDM=FLOAT(INDM)
      RDM=SQRT(FDM)
      IDR=IFIX(RDM)
      IDR2=2*IDR
   20 FORMAT(2X,'IDR =',I5,/)
C
C  J=ROW INDEX FOR OUTPUT ARRAY  I=COLUMN INDEX
C
      DO 300 J=1,IDR2
      IF(J.GT.IDR) GO TO 270
      DO 200 I=1,IDR
      IJ=IDR2*(J-1)+I
      IJK=IDR*(J-1)+I
      AROT(IJ)=ARIN(IJK)
  200 CONTINUE
C
C  FILL REMAINDER OF ROW WITH ZEROS
C
  220 DO 250 I=1,IDR
      IJ=IDR2*(J-1)+IDR+I
      AROT(IJ)=0.
  250 CONTINUE
      GO TO 300
  270 CONTINUE
C
C  FILL REMAINDER OF ROWS WITH ZEROS
C
      DO 280 I=1,IDR2
      IJ=IDR2*(J-1)+I
      AROT(IJ)=0.
  280 CONTINUE
  300 CONTINUE
      RETURN
      END
```

```
C
C   THIS SUBROUTINE CALCULATES THE CORRELATION BETWEEN
C   THE A VECTOR(DFT COEFFICIENTS) AND THE B VECTOR(DFT COEFFICIENTS)
C
      SUBROUTINE  CALCOR(KK2,JJ2,NV,AZ,EL,ROLL,J,N12,N22  )
      DIMENSION WH(16384),WHR(16384)
      DIMENSION S(128),INV(128),M(3)
      DIMENSION CMAX(20)
      DIMENSION MAXX(20),MAXY(20)
      DIMENSION W(128,128)
      COMPLEX*8 A(16384),B(16384)
      COMMON WHR,WH,A,CMAX
      M(1)=KK2
      M(2)=JJ2
      M(3)=0
 2037 FORMAT(/,5X,'J =',I2,3X,'CMAX =',F17.6,/)
  63  FORMAT(5X,'N12 =',I4,'N22 =',I4,'NV =',I4)
C
C     DOUBLE SIZE OF DATA ARRAY
C     WHR = INPUT VECTOR, WH  = OUTPUT VECTOR, NV = INPUT DIMENSION
C
      NV4=4*NV
      CALL VDOUBL(WHR,WH,NV,NV4)
C
C     PLOT DATA ARRAY
C
C     TRANSFER THE WH (K) DATA TO THE COMPLEX B MATRIX
C
      NDATA=0
      DO 104 I2=1,N22
      DO 103 I1=1,N12
      IWH= (I2-1)*N12+I1
      B(IWH)=WH (IWH)
      IF( WH(IWH) .GT.0.0) NDATA=NDATA+1
  103 CONTINUE
      IWHM=IWH-9
      WRITE(6,72) (WH(I),I=IWHM,IWH)
  104 CONTINUE
      WRITE(6,208) NDATA
  208 FORMAT('  NDATA =',I7)
      IF(NDATA.EQ.0) STOP
C
C     COMPUTE THE DISCRETE FOURIER TRANSFORM OF THE WH (K) DATA
C
   71 FORMAT(16(1X,F7.2))
   72 FORMAT (10(1X,E12.5))
   73 FORMAT(7(2X,I5))
      CALL HARM(B,M,INV,S,1,IPERR)
C
C                                          COMPLEX MULTIPLY
C
      DO 220 I1=1,NV4
```

```
      B(I1)=A(I1)*CONJG(B(I1))
      IF(I1 .LT.N22) WRITE(6,101) I1,A(I1),B(I1)
 220  CONTINUE
C
C                                                INVERSE FOURIER TRANSFORM
C
      CALL HARM(B,M,INV,S,-1,IPERR)
C
C     CALCULATE THE MAGNITUDE OF THE FOURIER COMPONENTS
C     AND STORE ONLY THE POS FREQ VALUES OF W(I1,I2)
C
      DO 30 I2=1,N22
      DO 30 I1=1,N12
      IWH= (I2-1)*N12+I1
      W(I1,I2)=CABS(B(IWH))
      IF(W(I1,I2).LT.CMAX(J)) GO TO 30
      CMAX(J)=W(I1,I2)
      MAXX(J)=I1
      MAXY(J)=I2
   30 CONTINUE
C
C     PRINT THE DFT COEFFICIENTS
C
      IF(MAXX(J).GT.3) GO TO 40
      MM3=1
      MM2=2
      MM1=3
      MAXX(J)=4
      MP1=5
      MP2=6
      MP3=7
      MP4=8
      GO TO 42
   40 CONTINUE
      N12M=N12-4
      IF(MAXX(J).GT.N12M) GO TO 41
      MM3=MAXX(J)-3
      MM2=MAXX(J)-2
      MM1=MAXX(J)-1
      MP1=MAXX(J)+1
      MP2=MAXX(J)+2
      MP3=MAXX(J)+3
      MP4=MAXX(J)+4
      GO TO 42
   41 CONTINUE
      MM3=N12-7
      MM2=N12-6
      MM1=N12-5
      MAXX(J)=N12-4
      MP1=N12-3
      MP2=N12-2
      MP3=N12-1
      MP4=N12
   42 CONTINUE
```

```
      WRITE(6,100) MM3,MM2,MM1,MAXX(J),MP1,MP2,MP3,MP4
100   FORMAT('      K',5X,'A(',I3,',K)',7(6X,'A(',I3,',K)'))
101   FORMAT(2X,I4,8(1X,E12.5))
      DO 61 I2=1,N22
      WRITE(6,101)I2,(W(I1,I2),I1=MM3,MP4)
61    CONTINUE
      RETURN
      END
```

```
C
C   KAL IS A RADAR RANGE ESTIMATOR USING A KALMAN FILTER.
C   THERE ARE 2 RANGE MEASUREMENTS AND A RADIAL ACCELERATION ESTIMATION
C   MEASUREMENT.
C
C  MODEL
C   X(K+1) = H(K)*X(K)+G(K)*U(K)
C   Y(K+1) = S(K+1)*X(K+1)+V(K+1)
C  KALMAN ESTIMATE
C   X'(K+1) = H(K)*X-HAT(K)
C   X-HAT(K+1) = X'(K)-K(K+1)*(S(K+1)*X'(K)-Y(K+1))
C   K(K+1) = P'(K+1)*ST(K+1)*(S(K+1)*P'(K+1)*ST(K+1)+R(K+1))**-1
C   P'(K+1) = H(K)*P(K)*HT(K)+G(K)*Q(K)*GT(K)
C   P(K+1) = P'(K+1)-K(K+1)*S(K+1)*P'(K+1)
C   P(0) = G(0)*Q(0)*GT(0)
C   T = TRANSPOSE
C   SUBROUTINES USED:  INTCON
C                      MANVR
C                      FTKALI
      DIMENSION X(4),H(4,4),G(4),S(3,4) ,P(4,4),T1(4,4),T2(4,4),T3(3),
     *TX(4),Y(3),R(3,3),F6(4,4)
      DIMENSION TP(100),EP1(100),XP1(100),RNOIS(100),VP1(100),VP2(100),
     *RNOIS2(100),XTP(100),ZTP(100)
      DIMENSION AP1(100),AP2(100),AP3(100)
      DIMENSION WHEN(2),TMEOD(2)
      DIMENSION PAUV(3),ANUV(3),ANL(3),ANLR(3),TPA(3)
      DIMENSION IOPT(5),STAT(5)
      DIMENSION IPAK(150)
      COMMON XT,YT,ZT,VXT,VYT,VZT,VELOC,T1ANG,T2ANG,DELR,IG,TRAT1,
     .DELT,NPRF,TRAT2,II,NPH,R90,IG2,ICTS,PI,IR,IH,IOS
      COMMON/INTC/   AZ,EL,ROL1,ROL2,ROL4,AZDEG,ELDEG,ROL1DG ,ROL2DG,
     *ROL4DG ,TGS1,TGS2,RT,RDOTT,R2DOTT,   DT,TRAD1,TRAD2,RANGT,
     *RANGTM,XR,YR,ZR,VXTM,VYTM,VZTM,VR2D,AXT,AYT,AZT,AXTG,AYTG,AZTG,
     *ANA,ANE,PAUV
      DATA IN,N,IT/3*4/
      DATA IS,M1/2*3/
      DATA IL,L/2*1/
      DATA ISEED/42573/
      DATA PI/3.141592654/
C   1/(S**2) MODEL STATE VECTOR
      CALL FR80ID(' 6220 R.H. WILLIAMS 6262 ')
      NI=5
      NO=6
      TGS1=10.
      TGS2=10.
      DT=0.05
      NP=31
      NPM=NP-1
      NPH=NP/10
      RES=0.5
      R90=PI/2.
      R270=3.*PI/2.
      GYD=10.7252
      NRUN=0
   10 CONTINUE
```

```
C
C   3-D TARGET TRAJECTORY DATA
C    ORIGIN AT INITIAL TARGET POSITION
C   RT = TRUE RANGE RELATIVE TO RADAR
C   RDOTT = TRUE VELOCITY RELATIVE TO RADAR
C   R2DOTT = TRUE ACCELERATION RELATIVE TO RADAR
C   VELOC = TARGET VELOCITY RELATIVE TO ORIGIN
C        TRAD- = TURN RADIUS OF TGS- G. TURN
C        TRAT- = TURN RATE   OF TGS- G. TURN  (RAD/SEC)
C        -R = RADAR LOCATION AT T(0)
C        -T = TRUE X,Y,Z TARGET POSITION
C        V-T = TRUE X,Y,Z VELOCITY
C        A-T = TRUE X,Y,Z ACCELERATION
C        A-TG = TRUE X,Y,Z ACCELERATION (G'S)
C   ANA = VERTICAL (AZ) ACCELERATION RELATIVE TO TARGET ORIENTATION
C   ANE = HORIZONTAL (EL) ACCELERATION RELATIVE TO TARGET ORIENTATION
C   PAUV(I) = PITCH AXIS UNIT VECTOR
C   PAUV(1) = X
C   PAUV(2) = Z
C   PAUV(3) = Y
C
C   ATC = TIME CONSTANT OF ACCELERATION
C   TCK = TRANSITION MATRIX ACCELERATION TIME CONSTANT
      ATC = 2.
      TC = ATC/DT
      TCK = (1.-EXP(-TC))
C   Q = COVARIANCE MATRIX OF U (LXL) - INPUT
C     = E[U*UT] = (RMS RANGE/SEC*SEC*SEC)**2
      READ(5,150) Q,RCV,ILAST
      DO 115 IOS=1,2
      IF(IOS.EQ.2) DT=0.1
      DO 118 IR=1,3
      DO 116 IH=1,2
C   H = STATE TRANSITION MATRIX (NXN)
      DATA H/16*0./
C   AGMTK = GAMMA ACCELERATION TRANSITION MATRIX CONSTANT
C   ARTK = NORMAL RADIAL LOAD ACCEL. TRANS. MATRIX CONSTANT
      IF(IH.EQ.1) AGMTK=1.0
      IF(IH.EQ.2) AGMTK=0.0
      IF(IH.EQ.3) AGMTK=0.5
      IF(IH.EQ.1) ARTK=0.0
      IF(IH.EQ.2) ARTK=1.0
      IF(IH.EQ.3) ARTK=0.5
      H(1,1)=1.
      H(1,2)=DT
      H(1,3)=AGMTK*DT*DT/2.
      H(1,4)=ARTK*DT*DT/2.
      H(2,2)=1.
      H(2,3)=DT*AGMTK
      H(2,4)=ARTK*DT
      H(3,3)=1.
C   ANLRK = RADIAL NORMAL LOAD ACCELERATION TRANSITION CONSTANT
      ANLRK=1.
      H(4,4)=ANLRK
C   G = STATE NOISE INPUT MATRIX (NXL)
```

```
  300 FORMAT('  Q =',F12.1,'  R =',F9.3,'  RT =',F10.1,'  RDOTT =',
     *4(F7.1),I3)
      CALL DATE(WHEN)
      CALL TOFDAY(TMEOD)
      WRITE(NO,112) WHEN(1),WHEN(2),TMEOD(1),TMEOD(2)
  112 FORMAT(2A6,2X,2A6)
      WRITE(6,90)
   90 FORMAT('   K',4X,'RT   ',7X,'Y',3X,
     *4X,'X(1)',7X,'X(2)',2X,'ERROR=RT-X(1)',2X,'NOISE',
     *2X,'AXTG',2X,'AYTG',2X,'AZTG',2X,'2ND DIFF-VELOC',/)
      WRITE(6,130) K,RT,Y(1),X(1),X(2),EP1( 1),RNOIS(1)
C
C  MAIN LOOP
C
      DO 100 II=1,NPM
      IP=II+1
      IPS=0
      IF(II.EQ.1) IPS=1
      PCTNS=0.7
      RNOIS(II)=SQRT(R(1,1))*GGNOF(ISEED)*PCTNS
      RNOIS2(II)=SQRT(R(2,2))*GGNOF(ISEED)*PCTNS
      DELR=VELOC*DELT*FLOAT(NPRF)
C  MANVR CALCULATES TRUE POSITIONS (XT,YT,ZT) AND TRUE VELOCITIES (VXT,
C  VYT,VZT) TARGET STARTS @ ORIGIN @ T'0, AND FLIES TOWARD RADAR FIXED
C   LOCATION RT ON X AXIS
C  X = DOWN RANGE, REFERENCED TO TARGET @ T =0
C  Y =  UP  RANGE, REFERENCED TO TARGET @ T =0
C  Z =  OFF RANGE, REFERENCED TO TARGET @ T =0
      CALL MANVR
C  V = TOTAL TRUE VELOCITY RELATIVE TO COORD SYSTEM , T(0)
      V=SQRT(VXT*VXT+VYT*VYT+VZT*VZT)
      AXT=(VXT-VXTM)/(DELT*FLOAT(NPRF))
      AYT=(VYT-VYTM)/(DELT*FLOAT(NPRF))
      AZT=(VZT-VZTM)/(DELT*FLOAT(NPRF))
C  AT = TOTAL TRUE ACCELERATION RELATIVE TO COORD SYSTEM , T(0)
      AT=SQRT(AXT*AXT+AYT*AYT+AZT*AZT)
      AXTG=AXT/10.7252
      AYTG=AYT/10.7252
      AZTG=AZT/10.7252
      ANE=((AYT*VXT-VYT*AXT)*VXT-(AZT*VYT-VZT*AYT)*VZT)/(V*VH)
      ANA=(AZT*VZT-VZT*AXT)/VH
C  AH = HORIZONTAL ACCELERATION - RELATIVE TO COORD. SYS., T(0)
      AH=SQRT(AXT*AXT+AZT*AZT)
      ELE=ANE+COS(EL)
      ELN=SQRT(ANA*ANA+ELE*ELE)
C  VH = HORIZONTAL VELOCITY       - RELATIVE TO COORD. SYS., T(0)
      VH=SQRT(VXT*VXT+VZT*VZT)
      AZ=ATAN2(VZT,VXT)
      AZDEG=AZ*180./PI
      EL=ATAN2(VYT,VH)
      ELDEG=EL*180./PI
      ANAEL=ANA/ELE
C  ICTS = COORDINATED TURN SWITCH
      IF(ICTS.EQ.1) ROL4=ATAN2(ANA,10.7252)
      ROL5 =ATAN2(ANA,10.7252)
```

```
      G(1)=DT*DT*DT/4.
      G(2)=DT*DT/2.
      G(3)=DT
      G(4)=    G(3)
C   R = COVARIANCE MATRIX OF V (M1XM1) - MEASUREMENT
      DATA R/9*0./
        R(1,1)=RCV
        R(2,2)=     R(1,1)
      R(3,3)=100*R(1,1)
C   S = OBSERVER MATRIX (M1XN)  Y = SX
      DATA S/12*0./
      S(1,1)=1.
      S(2,1)=1.
      S(3,4)=1.
C   INTCON INITILIZES TRAJECTORY VARIABLES
C
      CALL        INTCON
C   TX = ITERATED EXTENDED THRESHOLDS
      TX(1)=1.
      TX(2)=TX(1)
      TX(3)=TX(2)
      TX(4)=TX(3)
      K=0
      T=0.
      LL=0
      INCP=0
      RNOIS(1)=SQRT(R(1,1))*GGNOF(ISEED)
      RNOIS2( 1)=SQRT(R(2,2))*GGNOF(ISEED)
C   Y = INPUT OBSERVATION VECTOR (M1X1)
C   MULTIPLE MEASUREMENTS
      Y(1)=RT+RNOIS(1)
      Y(2)=RT+RNOIS2(1)
      Y(3)=0.
C   PRESET ESTIMATOR INITIAL STATE
C   X(1) = RANGE(K)
C   X(2) = RANGE RATE(K)
C   X(3) = RANGE ACCELERATION
C   X(4) = ESTIMATED RADIAL NORMAL LOAD ACCELERATION
      X100FF= 1.
      X20K=1.1
      X(1)=Y(1)+X100FF
      X(2)=X20K*RDOTT
      X(3)=0.
      X(4)=0.
      EP1( 1)= RT-X(1)
      XP1(1)=X(1)
      VP1(1)=X(2)
      VP2(1)=RDOTT
      AP1(1)=X(3)
      AP2(1)=0.
      AP3(1)=0.
      EPMAX=0.
      RDRX=RT
      NRUN=NRUN+1
      WRITE(6,300) Q,RCV,RT,RDOTT,AGMTK,ARTK,S(3,3),NRUN
```

```
  144 FORMAT('  COOD. TURN SW ='I3,'  ROL4 =',F9.3,'  COORD. TURN ROLL
     *',F9.3)
      ROLDEG=ROL4*180./PI
      ANADOT=((-VXT*AXT-VZT*AZT)*ANA)/(VH*VH)
      AZDOT=ANA/VH
      ELDOT=ANE/V
      ROLDOT=10.7252*ANADOT/(10.7252*10.7252+ANA*ANA)
C        V-TM = DELAYED TRUE X,Y,Z VELOCITY
      VXTM=VXT
      VYTM=VYT
      VZTM=VZT
C        -3 = X,Y,Z DIFFERENCE BETWEEN RADAR AND TRUE TARGET POSITION
      X3=XR-XT
      Y3=YR-YT
      Z3=ZR-ZT
C        RANGT = TRUE RANGE  RANGTM = DELAYED TRUE RANGE
      RANGTM=RANGT
      RANGT=SQRT(X3*X3+Y3*Y3+Z3*Z3)
      VNOIS=RNOIS(II)
      RANGN=RANGT+VNOIS
      RANGN2=RANGT+RNOIS2(II)
      REMDR=AMOD(RANGN,RES)
      REMDR2=AMOD(RANGN2,RES)
      RES2=RES/2.
C        RANGE = MEASURED RANGE WITH NOISE AND QUANTIZED TO RESOLUTION '
      RANGE=RANGN-REMDR
      RANGE2=RANGN2-REMDR2
      RT=RANGT
C   AMNL = MAGNITUDE OF NORMAL LOAD ACCELERATION ( G'S )
      AL=13.
      BE=-3.
      GAM=0.5
      EPS=GGNOF(ISEED)
      AMNL=AL+BE*EXP(GAM*EPS)
C   AZ =  AZDEG
C   EL = -ELDEG
C   ROLL = -ROLDEG
C   PAUV(1) = X
C   PAUV(2) = Z
C   PAUV(3) = Y
      SAZ=SIN( AZ)
      CAZ=COS(  AZ )
      SEL=SIN(-EL)
      CEL=COS(- EL )
      SRO=SIN(-ROL4)
      CRO=COS(-ROL4)
      TPA(1)=PAUV(1)*CEL*CAZ-PAUV(2)*CEL*SAZ+PAUV(3)*SEL
      TPA(2)=PAUV(1)*(CRO*SAZ+SRO*SEL*CAZ)+PAUV(2)*(CRO*CAZ-SRO*SEL*SA
     .-PAUV(3)*SRO*CEL
      TPA(3)=PAUV(1)*(SRO*SAZ-CRO*SEL*CAZ)+PAUV(2)*(CRO*SEL*SAZ+SRO*CA
     .+PAUV(3)*CRO*CEL
C   ANUV(I) = NORMAL ACCELERATION UNIT VECTOR
C           = (TPA(I) X V-T)/V    (VECTOR CROSS PRODUCT)
C             (X,Z,Y)
      ANUV(1)=((TPA(2)*VYT) - (TPA(3)*VZT))/V
```

```
      ANUV(2)=((TPA(3)*VXT) - (TPA(1)*VYT))/V
      ANUV(3)=((TPA(1)*VZT) - (TPA(2)*VXT))/V
C   ANL(I) = NORMAL LOAD ACCELERATION
      ANL(1) = ANUV(1)*AMNL
      ANL(2) = ANUV(2)*AMNL
      ANL(3) = ANUV(3)*AMNL
C   ANLR(I) = RADIAL NORMAL LOAD ACCELERATION( G'S )
C           = PROJECTION (DOT SCALAR PRODUCT) OF ANL(I) ON THE RADIAL
C             VECTOR FROM TARGET TO RADAR/RADIAL RANGE
      ANLR(1)=((ANL(1)*X3)+(ANL(2)*Z3)+(ANL(3)*Y3))/RANGT
      Y(1)=RANGE
      Y(2)=RANGE2
      Y(3)=ANLR(1)*GYD
C   VR2D = 1ST DIFFERANCE, RANGE VELOCITY  VR2DD = DELAYED VR2D
      VR2DD=VR2D
      VR2D=(RANGT-RANGTM)/(DELT*FLOAT(NPRF))
C   AR = RANGE  ACCELERATION, 2ND DIFFERANCE
      ARM=AR
      AR=(VR2D-VR2DD)/(DELT*FLOAT(NPRF))
                   WRITE(6,133) RANGT,RANGTM,VR2DD,VR2D,AR,Y(3),X(3),
     *X3,Y3,Z3
C   TEST RADIAL 2ND DIFFERENCE: AR, & LIMIT TO MAX. EXPECTED
C   RATE OF CHANGE: SQRT(Q)
      ADIF=ABS(AR-ARM)
      QLIM=SQRT(Q)*DT
      IF(ADIF.GT.QLIM)  AR=SIGN(QLIM,AR)+ARM
C   TEST FOR RE-INITILIZATION OF KALMAN FILTER
      SIGTH=2.0
      SIG1=SIGTH*SQRT(R(1,1))
      SIG2=SIGTH*SQRT(R(2,2))
      SIG3=AMAX1(SIG1,SIG2)
      PMAX=0.
      EEST=((Y(1)+Y(2))/2.)-X(1)
      IF(EEST   .LT.SIG3) GO TO 44
      INCP=INCP+1
      IPS=1
      DO 42 JP=1,N
        PMAX=AMAX1(P(JP,JP),PMAX)
   42 CONTINUE
      PPCT=1.0
      DO 43 JP=1,N
        P(JP,JP)=PMAX*PPCT
   43 CONTINUE
   44 CONTINUE
      WRITE(6,133) AZ,EL,ROL4,TPA(1),TPA(2),TPA(3),ANUV(1),ANUV(2),
     .ANUV(3),AMNL,ANLR(1),AR
      CALL FTKALI(K,X,H,G,Y,S,Q,R  ,P,IN,IS,IL,N,M1,L,T1,T2,IT,T3,
     *TX,FG,IER,IPS)
      T=T+DT
      EP1(IP)= RT-X(1)
      XP1(IP)=X(1)
      VP1(IP)=X(2)
      VP2(IP)=VR2D
      AP1(IP)=X(3)
      AP2(IP)=AR
```

```
      AP3(IP)=Y(3)
      TP(IP)=T
      XTP(IP)=XT
      ZTP(IP)=ZT
      VPCT=X(2)*DT*100./EP1(IP)
      APCT=X(3)*DT*DT*50./EP1(IP)
      PCTM=T2(1,1)*100./EP1(IP)
      WRITE(6,134) II,VPCT,APCT,PCTM,T2(1,1),PMAX
  134 FORMAT(1X,I3,' PCT VELOV, ACC, MEAS =',10(F12.3))
      IF(ABS(EP1(IP)).GT.ABS(EPMAX)) EPMAX=EP1(IP)
      IF(ABS(EP1(IP)).GT.40.) LL=LL+1
      K=K+1
      IF(FG(1,1).GE.1.) GO TO 51
      WNS=FG(2,1)/((1.-FG(1,1))*DT)
      IF(WNS.LT.0.) GO TO 51
      WN=SQRT(WNS)
      WNHZ=WN/2.*PI
      DR=(FG(1,1)-DT*FG(2,1))/(2*(1.-FG(1,1))*DT*WN)
  194 FORMAT('  NATURAL FREQ. (HZ) =',F9.2,'  DAMPING RATIO =',F8.3)
   51 CONTINUE
      WRITE(6,130) K,RT,Y(1),X(1),X(2),EP1(IP),RNOIS(II),AXTG,AYTG,AZT
     *VR2D
  100 CONTINUE
C
      IOPT(1)=1
      IOPT(5)=1
      NPP=NP+1
      CALL BEIUGR(EP1,NPP,IOPT,STAT,IER)
      WRITE(6,102) STAT(1),STAT(5),EPMAX
  102 FORMAT(  '  ARITHMETIC MEAN OF RANGE ERR =',F7.3,
     *';  VARIANCE (AVE. RMS ERROR) OF RANGE ERROR =',F9.3,
     *';  MAXIMUM RANGE ERR =',F9.3,/)
      WRITE(6,104) LL,INCP
  104 FORMAT(/,'  NO. OF RETURNS OUT OF GATE =',I4,
     *' NO. TIMES P INCR. =',I4)
      IRPLOT=1
      IF(IRPLOT.EQ.0) GO TO 113
      CALL INTAXS
      CALL YAXANG(0.)
      CALL SIMPLX
      CALL TITLE('          KALMAN FILTERS',100,
     *                    'ERROR IN YARDS$',
     . 100,'TIME IN SEC.$',100,6.,8.)
      CALL XTICKS(5)
      CALL YTICKS(5)
      XORG=AMIN1(ARMIN(EP1,NP),ARMIN(RNOIS,NP))
      XMAXG=AMAX1(ARMAX(EP1,NP),ARMAX(RNOIS,NP))
      CALL GRAF(XORG,'SCALE',XMAXG,
     *              ARMIN(TP   ,NP),'SCALE',ARMAX(TP   ,NP))
      CALL CURVE(EP1,  TP,NP,10)
      CALL CURVE(RNOIS,TP,NP,10)
      CALL MESSAG('COV Q =$',100,1., 8.9)
      CALL REALNO(    Q,0,'ABUT','ABUT')
      CALL MESSAG('  COV R =$',100,   'ABUT','ABUT')
      CALL REALNO(R(1,1),2.'ABUT','ABUT')
```

```
      CALL MESSAG(' # =$',      100,    'ABUT','ABUT')
      CALL INTNO(NRUN,'ABUT','ABUT')
      CALL DATE(WHEN)
      CALL TOFDAY(TMEOD)
      CALL MESSAG(WHEN,10,1.,9.3)
      CALL MESSAG(TMEOD,10,'ABUT','ABUT')
      IDUMMY=LINEST(IPAK,150,80)
      CALL LINES('ERRORS',IPAK,1)
      CALL LINES('NOISES',IPAK,2)
      CALL LEGEND(IPAK ,2,4.0,2.0)
      CALL ENDPL(0)
113   CONTINUE
      IVPLOT=0
      IF(IR .EQ.3) IVPLOT=1
      IF(IVPLOT.FQ.0) GO TO 114
      CALL INTAXS
      CALL YAXANG(0.)
      CALL SIMPLX
      CALL TITLE('          KALMAN FILTERS',100,
     *                  '    VELOC IN YARDS/SECS',
     . 100,'TIME IN SEC.$',100,6.,8.)
      CALL XTICKS(5)
      CALL YTICKS(5)
      XORG=AMIN1(ARMIN(VP1,NP),ARMIN(  VP2,NP))
      XMAXG=AMAX1(ARMAX(VP1,NP),ARMAX(  VP2,NP))
      CALL GRAF(XORG,'SCALE',XMAXG,
     *              ARMIN(TP  ,NP),'SCALE',ARMAX(TP   ,NP))
      CALL CURVE(VP1,  TP,NP,10)
      CALL CURVE(VP2,TP,NP,10)
      CALL MESSAG('COV Q =$',100,1., 8.9)
      CALL REALNO(    Q,0,'ABUT','ABUT')
      CALL MESSAG('  COV R =$',100,   'ABUT','ABUT')
      CALL REALNO(R(1,1),2,'ABUT','ABUT')
      CALL MESSAG(' # =$',      100,    'ABUT','ABUT')
      CALL INTNO(NRUN,'ABUT','ABUT')
      CALL DATE(WHEN)
      CALL TOFDAY(TMEOD)
      CALL MESSAG(WHEN,10,1.,9.3)
      CALL MESSAG(TMEOD,10,'ABUT','ABUT')
      IDUMMY=LINEST(IPAK,150,80)
      CALL LINES('X(2)$',IPAK,1)
      CALL LINES('VR2D$',IPAK,2)
      CALL LEGEND(IPAK ,2,5.0,1.0)
      CALL ENDPL(0)
114   CONTINUE
      IAPLOT=1
      IF(IAPLOT.EQ.0) GO TO 1148
      CALL INTAXS
      CALL YAXANG(0.)
      CALL SIMPLX
      CALL TITLE('          KALMAN FILTERS',100,
     *                  ' ACCEL IN YDS/SEC*SECS',
     . 100,'TIME IN SEC.$',100,6.,8.)
      CALL XTICKS(5)
      CALL YTICKS(5)
```

```
      XORG=AMIN1(ARMIN(AP1,NP),ARMIN(   AP2,NP))
      XMAXG=AMAX1(ARMAX(AP1,NP),ARMAX(   AP2,NP))
      CALL GRAF(XORG,'SCALE',XMAXG,
     *           ARMIN(TP   ,NP),'SCALE',ARMAX(TP   ,NP))
      CALL CURVE(AP1,  TP,NP,10)
      CALL CURVE(AP2,TP,NP,10)
      CALL CURVE(AP3,TP,NP,10)
      CALL MESSAG('COV Q =$',100,1., 8.9)
      CALL REALNO(    Q,0,'ABUT','ABUT')
      CALL MESSAG('  COV R =$',100,   'ABUT','ABUT')
      CALL REALNO(R(1,1),2,'ABUT','ABUT')
      CALL MESSAG(' # =$',     100,   'ABUT','ABUT')
      CALL INTNO(NRUN,'ABUT','ABUT')
      CALL DATE(WHEN)
      CALL TOFDAY(TMEOD)
      CALL MESSAG(WHEN,10,1.,9.3)
      CALL MESSAG(TMEOD,10,'ABUT','ABUT')
      IDUMMY=LINEST(IPAK,150,80)
      CALL LINES('X(3)$',IPAK,1)
      CALL LINES('AR  $',IPAK,2)
      CALL LINES('Y(3)$',IPAK,3)
      CALL LEGEND(IPAK ,3,4.0,2.0)
      CALL ENDPL(0)
 1148 CONTINUE
      CALL INTAXS
      CALL YAXANG(0.)
      CALL SIMPLX
      CALL TITLE('TARGET GROUND PLOT   $',100,
     *' OFF RANGE (Z) - YDS.$',100,
     *' DOWN RANGE (X) - YDS.$',100,6.,8.)
      CALL XTICKS(5)
      CALL YTICKS(5)
      XORG=      ARMIN(ZTP,NP)
      XMAXG=      ARMAX(ZTP,NP)
      YMAXG=ARMAX(XTP,NP)
      CALL GRAF(XORG,'SCALE',XMAXG,
     *           ARMIN(XTP ,NP),'SCALE',YMAXG)
      NPLS= 5
      CALL CURVE(ZTP,XTP,NP,NPLS)
C   DRAW + AT RADAR LOCATION, XT=RT , ZT=0.
      CALL MARKER(3)
      CALL SCLPIC(2.)
      RDRY=0.
      IF(RDRX.LE.YMAXG) CALL CURVE(RDRY,RDRX,1,1)
      CALL RESET('SCLPIC')
      CALL RESET('MARKER')
      CALL MESSAG('COV Q =$',100,1., 8.9)
      CALL REALNO(    Q,0,'ABUT','ABUT')
      CALL MESSAG('  COV R =$',100,   'ABUT','ABUT')
      CALL REALNO(R(1,1),2,'ABUT','ABUT')
      CALL MESSAG(' # =$',     100,   'ABUT','ABUT')
      CALL INTNO(NRUN,'ABUT','ABUT')
      CALL DATE(WHEN)
      CALL TOFDAY(TMEOD)
      CALL MESSAG(WHEN,10,1.,9.3)
```

```
      CALL MESSAG(TMEOD,10,'ABUT','ABUT')
      CALL ENDPL(0)
  116 CONTINUE
  118 CONTINUE
  115 CONTINUE
      IF(ILAST.NE.1) GO TO 10
      CALL DONEPL
  120 FORMAT(1X,I4,1X,F6.1,3(F6.2),6(F9.3),6(F7.3))
  130 FORMAT  (1X,I3, 4(F10.1),7(F10.3))
  133 FORMAT(1X,14F9.2)
  150 FORMAT ()
      END
```

```
      SUBROUTINE INTCON
C   INTCON INITILIZES TRAJECTORY VARIABLES
      DIMENSION PAUV(3)
      COMMON XT,YT,ZT,VXT,VYT,VZT,VELOC,T1ANG,T2ANG,DELR,IG,TRAT1,
     .DELT,NPRF,TRAT2,II,NPH,R90,IG2,ICTS,PI,IR,IH,IOS
      COMMON/INTC/    AZ,EL,ROL1,ROL2,ROL4,AZDEG,ELDEG,ROL1DG ,ROL2DG,
     *ROL4DG ,TGS1,TGS2,RT,RDOTT,R2DOTT,    DT,TRAD1,TRAD2,RANGT,
     *RANGTM,XR,YR,ZR,VXTM,VYTM,VZTM,VR2D,AXT,AYT,AZT,AXTG,AYTG,AZTG,
     *ANA,ANE,PAUV
      ITCS=1
      AZ=0.
      EL=0.
      ROL1=0.
      ROL2=0.
      ROL4=0.
      AZDEG=0.
      ELDEG=0.
      ROL1DG=0.
      ROL2DG=0.
      T1ANG=0.
      IG=0
      IG2=0
      DELT=DT
      NPRF=1
C   3-D TARGET TRAJECTORY DATA
C    ORIGIN AT INITIAL TARGET POSITION
C   RT = TRUE RANGE RELATIVE TO RADAR
C   RDOTT = TRUE VELOCITY RELATIVE TO RADAR
C   R2DOTT = TRUE ACCELERATION RELATIVE TO RADAR
      IF(IR.EQ.1) RT=11275.
      IF(IR.EQ.2) RT=1000.
      IF(IR.EQ.3) RT=150.
      IF(IOS.EQ.1) RDOTT=-175.
      IF(IOS.EQ.2) RDOTT=-100.
      R2DOTT=0.
C   VELOC = TARGET VELOCITY RELATIVE TO ORIGIN
      VELOC=-RDOTT
C         TRAD- = TURN RADIUS OF TGS- G. TURN
      TRAD1=VELOC*VELOC/TGS1*10.7252
      TRAD2=VELOC*VELOC/TGS2*10.7252
C         TRAT- = TURN RATE   OF TGS- G. TURN  (RAD/SEC)
      TRAT1=TGS1*10.7252/VELOC
      TRAT2=TGS2*10.7252/VELOC
      RANGT=RT
      RANGTM=RANGT-(RDOTT*DELT*FLOAT(NPRF))
C         -R = RADAR LOCATION AT T(0)
      XR=RT
      YR=0.
      ZR=0.
C         -T = TRUE X,Y,Z TARGET POSITION
      XT=0.
      YT=0.
      ZT=0.
C         V-T = TRUE X,Y,Z VELOCITY
      VXT=VELOC
      VYT=0.
      VZT=0.
      VXTM=VXT
      VYTM=VYT
      VZTM=VZT
```

```
      VR2D=RDOTT
C        A-T = TRUE X,Y,Z ACCELERATION
      AXT=0.
      AYT=0.
      AZT=0.
C        A-TG = TRUE X,Y,Z ACCELERATION (G'S)
      AXTG=0.
      AYTG=0.
      AZTG=0.
C   ANA = VERTICAL (AZ) ACCELERATION RELATIVE TO TARGET ORIENTATION
C   ANE = HORIZONTAL (EL) ACCELERATION RELATIVE TO TARGET ORIENTATION
      ANA=0.
      ANE=0.
C   PAUV(I) = PITCH AXIS UNIT VECTOR
C   PAUV(1) = X
C   PAUV(2) = Z
C   PAUV(3) = Y
      PAUV(1)=0.
      PAUV(2)=1.
      PAUV(3)=0.
      END
```

```
    FLECS,S    MVR,R.MVRR,MVRT

-----------------------------------------

MANVR CALCULATES TRUE POSITIONS (XT,YT,ZT) AND TRUE VELOCITIES (VXT,
VYT,VZT) TARGET STARTS @ ORIGIN @ T'O, AND FLIES TOWARD RADAR FIXED @
 LOCATION RT ON X AXIS
X = DOWN RANGE, REFERENCED TO TARGET @ T =0
Y =  UP  RANGE, REFERENCED TO TARGET @ T =0
Z =  OFF RANGE, REFERENCED TO TARGET @ T =0
    SUBROUTINE MANVR
    COMMON XT,YT,ZT,VXT,VYT,VZT,VELOC,T1ANG,T2ANG,DELR,IG,TRAT1,
   .DELT,NPRF,TRAT2,II,NPH,R90,IG2,ICTS,PI,IP,IH,IOS
    COMMON/INTC/    AZ,EL,ROL1,ROL2,ROL4,AZDEG,ELDEG,ROL1DG ,ROL2DG,
   *ROL4DG ,TGS1,TGS2,RT,RDOTT,R2DOTT,    DT,TRAD1,TRAD2,RANGT,
   *RANGTM,XR,YR,ZR,VXTM,VYTM,VZTM,VR2D,AXT,AYT,AZT,AXTG,AYTG,AZTG,
   *ANA,ANE,PAUV
*ROL4 ALONG X-AXIS ROTATES PITCH VECTOR CLOCKWISE. (FROM X,Y,Z =
 0,0,1  TO  0,-A,B )
RR = ROLL RATE OF AIRCRAFT (DEG/SEC)
RRR = ROLL RATE OF AIRCRAFT (RAD/SEC)
ICTS = COORDINATED TURN SWITCH
    DT=DELT
    RR=85.
    RRR=RR*PI/180.
    CONDITIONAL
    .  (II.LT.NPH)
    .  .  INITIAL TRAJECTORY
    .  .  ITCS=0
    .  .  XT=XT+DELR
    .  .  'MEASURED' AIRCRAFT ROLL, PRECEEDING MANUVER
    .  .  ROL4=ROL4+(RRR*DT)
    .  ...FIN
    .  (T1ANG.LT.R90)
    .  .  SECOND  TRAJECTORY
    .  .  ROL4=ROL4+(RRR*DT)
    .  .  IF(ROL4.GT.1.55)
    .  .  .  ROL4=1.55
    .  .  ...FIN
    .  .  IF(IG.GT.18)
    .  .  .  ICTS=0
    .  .  .  'MEASURED' AIRCRAFT ROLL, PRECEEDING MANUVER
    .  .  .  ROL4=ROL4-(RRR*DT)
    .  .  ...FIN
    .  .  IG=IG+1
    .  .    T1ANG = AZ ANGLE OF TARGET
    .  .  T1ANG=TRAT1*DELT*FLOAT(NPRF)*IG
    .  .  XT=XT+DELR*COS(T1ANG)
    .  .  ZT=ZT+DELR*SIN(T1ANG)
    .  .   @R90  VXT = 0,  VZT = VELOC
    .  .  VXT=VELOC*COS(T1ANG)
    .  .  VZT=VELOC*SIN(T1ANG)
    .  ...FIN
    .  (.TRUE.)
```

```
.   .   THIRD   TRAJECTORY
.   .   'MEASURED' AIRCRAFT ROLL
.   .   IF(ROL4.GT.0.0)
.   .   .  ROL4=ROL4-(RRR*DT)
.   .   ...FIN
.   .   IF(ROL4.LT.0.0)
.   .   .  ROL4=0.0
.   .   ...FIN
.   .   IG2=IG2+1
.   .   T2ANG=TRAT2*DELT*FLOAT(NPRF)*IG2
.   .   YT=YT+DELR*SIN(T2ANG)
.   .   ZT=ZT+DELR*COS(T2ANG)
.   .   VYT=VELOC*SIN(T2ANG)
.   .   VZT=VELOC*COS(T2ANG)
.   ...FIN
...FIN
RETURN
END

CS VERSION P22.6)

------------------------------------
```

```
C        SUBROUTINE FTKALI (K,X,H,G,Y,S,Q,R,P,IN,IS,IL,N,M1,L,T1,T2,
C-FTKALI--------S-------------------------------------------------------
C     *                          IT,T3,TX,IER)
C
C    FUNCTION               - ITERATED        KALMAN FILTERING
C    USAGE                  - CALL FTKAIE(K,X,H,G,Y,S,Q,R,P,IN,IS,IL,N,M1,L
C                               T1,T2,IT,T3,TX,IER)
C    PARAMETERS   K         - INPUT STEP COUNTER. K=0,1,2,... WHEN K=0,
C                               VECTOR X SHOULD CONTAIN THE PRIOR ESTIMATE
C                               OF THE MEAN OF X, AND THE PROGRAM
C                               CALCULATES THE ESTIMATED VARIANCE OF
C                               X AS P=GQG-TRANSPOSE AT STEP 0.
C                 X         - INPUT/OUTPUT VECTOR OF LENGTH N. ON INPUT,
C                               X IS THE STATE VECTOR AT STEP K, AND ON
C                               OUTPUT, X CONTAINS THE ESTIMATED STATE
C                               VECTOR AT STEP K+1.
C                 H         - INPUT MATRIX OF DIMENSION N BY N. H IS THE
C                               TRANSITION MATRIX AT STEP K.
C                 G         - INPUT MATRIX OF DIMENSION N BY L AT STEP K.
C                 Y         - INPUT OBSERVATION VECTOR OF LENGTH M1 AT
C                               STEP K+1.
C                 S         - INPUT MATRIX OF DIMENSION M1 BY N AT STEP K+1
C                 Q         - INPUT COVARIANCE MATRIX OF U, OF DIMENSION L
C                               AT STEP K.
C                 R         - INPUT COVARIANCE MATRIX OF V, OF DIMENSION
C                               M1 BY M1 AT STEP K+1.
C                 P         - INPUT/OUTPUT MATRIX OF DIMENSION N BY N.
C                               ON INPUT, P IS THE VARIANCE MATRIX OF X
C                               AT STEP K.  ON OUTPUT, P IS THE ESTIMATED
C                               VARIANCE MATRIX OF X AT STEP K+1.
C                 IN        - FIRST DIMENSION OF H,G, AND P AS DIMENSIONED
C                               IN CALLING PROGRAM.
C                 IS        - FIRST DIMENSION OF S AND R AS DIMENSIONED IN
C                               CALLING PROGRAM.
C                 IL        - FIRST DIMENSION OF Q AS DIMENSIONED IN
C                               CALLING PROGRAM.
C                 N         - INPUT. SEE DESCRIPTIONS OF X,H,G,M,P.
C                               N MUST BE GREATER THAN 0.
C                 M1        - INPUT. SEE DESCRIPTIONS OF Y,S,R,T3.
C                               M1 MUST BE GREATER THAN 0.
C                 L         - INPUT. SEE DESCRIPTIONS OF G,Q.
C                               L MUST BE GREATER THAN 0.
C                 T1        - WORK MATRIX OF DIMENSION NM BY NM, WHERE
C                               NM IS THE MAXIMUM OF N AND M1.
C                 T2        - WORK MATRIX OF DIMENSION NM BY NML, WHERE
C                               NML IS THE MAXIMUM OF NM AND L.
C                 IT        - FIRST DIMENSION OF T1 AND T2 AS
C                               DIMENSIONED IN CALLING PROGRAM.
C                 T3        - WORK VECTOR OF LENGTH M1.
C                 TX        - INPUT. ITERATED EXTENDED THRESHOLD
C                 IER       - ERROR PARAMETER.  IER = 0 IMPLIES NO ERROR.
C                           TERMINAL ERROR = 128+N
C                               N = 1 INDICATES ONE OF IN, IS, IL, OR IT
C                                 IS TOO SMALL, OR THAT ONE OF N, M1,
C                                 OR L IS NOT A POSITIVE INTEGER.
```

```
C                                      N = 2 INDICATES MATRIX INVERSION FAILED.
C    PRECISION             - SINGLE
C    REQ'D IMSL ROUTINES   - LEQT1F,LUDATF,LUELMF,UERTST,VMULFF,VMULFP
C    LANGUAGE              - FORTRAN
C-----------------------------------------------------------------------
C    LATEST REVISION       - MAY 10, 1979
C MODEL
C   X(K+1) = H(K)*X(K)+G(K)*U(K)
C   Y(K+1) = S(K+1)*X(K+1)+V(K+1)
C
      SUBROUTINE FTKALI (K,X,H,G,Y,S,Q,R,P,IN,IS,IL,
     *                   N,M1,L,T1,T2,IT,T3,TX,FG,IER,IPS)
C
      DIMENSION          X(1),H(IN,1),G(IN,1),Y(1),S(IS,1),Q(IL,1),
     *                   R(IS,1),P(IN,1),T1(IT,1),T2(IT,1),T3(1)
      DIMENSION XK(20),        XP(20),EX(20),TX(IN),T4(20),HNL(20),
     *EXM(20),FG(IT,1)
      DIMENSION T5(20),T6(10,10)
      DATA               I0/0/,I1/1/,I20/20/
      IF (IN.GE.N .AND. IS.GE.M1 .AND. IL.GE.L
     *    .AND. (IT.GE.N .OR. IT.GE.M1) .AND. N.GT.0
     *    .AND. M1.GT.0 .AND. L.GT.0) GO TO 5
      IER = 129
      GO TO 9000
    5 IER = 0
C                                      CALCULATE P IF K = ZERO
C   P(0) = G(0)*Q(0)*GT(0)
      IF (K .NE. 0) GO TO 10
      CALL VMULFF ( G, Q, N, L, L,IN,IL,T2,IT,IER)
      CALL VMULFP (T2, G, N, L, N,IT,IN, P,IN,IER)
      IF(IPS.EQ.0) GO TO 166
      WRITE(6,160) (P(1,J),J=1,IN),(TX(I),I=1,IN)
      DO 165 I=2,IN
      WRITE(6,161) (P(I,J),J=1,IN)
  165 CONTINUE
  166 CONTINUE
  160 FORMAT('  P(0) =',10(E10.4))
  161 FORMAT(8X,10(E10.4))
   10 CONTINUE
C   SAVE INITIAL X
      DO 12 I=1,N
      XK(I)=X(I)
   12 CONTINUE
C                                      CALCULATE X-PRIME AT STEP K+1
C   X'(K+1) = H(K)*X-HAT(K)
      CALL VMULFF (H, X, N, N,I1,IN,IN,T1,IT,IER)
      DO 15 I = 1,N
         XP(I) = T1(I,1)
         X(I)=XP(I)
   15 CONTINUE
      IF(IPS.EQ.1) WRITE(6,170) (XP(I),I=1,N)
  170 FORMAT('  X-PRIME = H.X =',10(F10.2))
C                                      CALCULATE P-PRIME AT STEP K+1
C   P'(K+1) = H(K)*P(K)*HT(K)+G(K)*Q(K)*GT(K)
      CALL VMULFF ( H, P, N, N, N,IN,IN,T1,IT,IER)
```

```
      CALL VMULFP (T1, H, N, N, N,IT,IN, P,IN,IER)
      CALL VMULFF ( G, Q, N, L, L,IN,IL,T2,IT,IER)
      CALL VMULFP (T2, G, N, L, N,IT,IN,T1,IT,IER)
      DO 25 I = 1,N
         DO 20 J = 1,N
            P(I,J) = P(I,J) + T1(I,J)
   20    CONTINUE
   25 CONTINUE
      IF(IPS.EQ.0) GO TO 186
      WRITE(6,180) (P(1,J),J=1,IN)
      DO 185 I=2,IN
      WRITE(6,181) (P(I,J),J=1,IN)
  185 CONTINUE
  186 CONTINUE
  180 FORMAT('  P-PRIME =',10(E10.4))
  181 FORMAT(11X,10(F10.4))
C                                        CALCULATE MATRIX K AT STEP K+1
C   K(K+1) = P'(K+1)*ST(K+1)*(S(K+1)*P'(K+1)*ST(K+1)+R(K+1))**-1
      CALL VMULFP ( S, P,M1, N, N,IS,IN,T2,IT,IER)
      IF(IPS.EQ.0) GO TO 28
      DO 27 I=1,M1
      WRITE(6,187) I,(T2(I,JJ),JJ=1,M1)
   27 CONTINUE
   28 CONTINUE
      CALL VMULFP (T2, S,M1, N,M1,IT,IS,T1,IT,IER)
      DO 35 I = 1,M1
         DO 30 J = 1,M1
            T1(I,J) = T1(I,J) + R(J,I)
   30    CONTINUE
      IF(IPS.EQ.1) WRITE(6,187) I,(T1(I,JJ),JJ=1,M1)
   35 CONTINUE
  187 FORMAT('  S.P.ST+R(',I1,',J) = T1 =',10(F10.2))
      CALL LEQT1F(T1, N,M1,IT,T2,IO,T3,IER)
      IF (IER .EQ. 0) GO TO 40
      IER = 130
      GO TO 9000
   40 DO 50 I = 1,M1
         DO 45 J = 1,N
            T1(J,I) = T2(I,J)
            FG(J,I)=T1(J,I)
   45    CONTINUE
   50 CONTINUE
      IF(IPS.EQ.0) GO TO 194
      WRITE(6,190) (T1(1,I),I=1,M1)
      DO 193 I=2,IN
      WRITE(6,191) (T1(I,J),J=1,M1)
  193 CONTINUE
  194 CONTINUE
  190 FORMAT('  K = T1 =',10(F10.2))
  191 FORMAT(10X,10(F10.2))
C                                        CALCULATE X-HAT AT STEP K+1
C KALMAN ESTIMATE
C  X-HAT(K+1) = X'(K)-K(K+1)*(S(K+1)*X'(K)-Y(K+1))
      KL=0
       DO 53 I=1,IN
```

```
      EX(I)=0.
      EXM(I)=1E9
   53 CONTINUE
   54 CONTINUE
      DO 542 I=1,IN
        T4(I)=XP(I)-X(I)
  542 CONTINUE
C  OBS. ERR. = S(X'-X)-Y+HNL = T5
   55 CALL VMULFF ( S, X,M1, N,I1,IS,IN,T3,IS,IER)
      CALL VMULFF ( S,T4,M1, N,I1,IS,IN,T5,IS,IER)
      CALL VMULFF( S, X,M1, N,I1,IS,IN,HNL,I20,IER)
      DO 60 I = 1,M1
         T3(I) = T3(I) - Y(I)
   60 CONTINUE
      IF(IPS.EQ.1) WRITE(6,195) (T3(I),I=1,M1),(X(I),I=1,N)
  195 FORMAT('  OBS ERR = S.X-Y =',12(F9.2))
C  T1 = GAIN K
      CALL VMULFF (T1,T3, N,M1,I1,IT,IS,T2,IT,IER)
      DO 65 I = 1,N
         X(I) = XP(I) - T2(I,1)
      T6(I,1)=T2(I,1)
   65 CONTINUE
      IF(IPS.EQ.1) WRITE(6,200) (X(I),I=1,N),(T2(I,1),I=1,N)
  200 FORMAT('  X-HAT =',10(F10.2))
   69 CONTINUE
C                                         CALCULATE P AT STEP K+1
C  P(K+1) = P'(K+1)-K(K+1)*S(K+1)*P'(K+1)
      CALL VMULFF (T1, S, N,M1, N,IT,IS,T2,IT,IER)
      CALL VMULFF (T2, P, N, N, N,IT,IN,T1,IT,IER)
      DO 75 I = 1,N
      T2(I,1)=T6(I,1)
         DO 70 J = 1,N
            P(I,J) = P(I,J) - T1(I,J)
   70    CONTINUE
   75 CONTINUE
      IF(IPS.EQ.0) GO TO 216
      WRITE(6,210) (P(1,J),J=1,IN)
      DO 215 I=2,IN
      WRITE(6,211) (P(I,J),J=1,IN)
  215 CONTINUE
  216 CONTINUE
      GO TO 9005
 9000 CONTINUE
      CALL UERTST (IER,6HFTKALM)
 9005 RETURN
  120 FORMAT(1X,I4,1X,F6.1,10(F11.2))
  130 FORMAT(1X,I3,12(F10.3))
  140 FORMAT(2X,I3,12(F10.2))
  210 FORMAT('  P =',10(F10.2))
  211 FORMAT( 5X,10(F10.2))
      END
```

BIBLIOGRAPHY

1. By: Infared Countermeasures Branch, Infared Signatures: Theory, Techniques, and Equipment, NWC Tech. Publ. 4902, 1971.

2. S. A. Dudani, An Experimental Study of Moment Methods for Automatic Identification of Three Dimensional Objects from TV Images, Ohio St. Univ., PhD. Diss., 1973.

3. R. L. Pio, "Euler Angle Transforms", IEEE Trans. on Auto. Control, Vol. AC-11, no. 4, pp. 707-715, Oct. 1966.

4. E. H. Ehling, Range Instrumentation, Prentice-Hall Inc., Englewood Cliffs, N. J., 1967.

5. A. V. Oppenheim and R. W. Schafer, Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1975.

6. W. C. Effinger, Fortran Subroutine to Calculate Projected Surface Areas of Rotated Three Dimensional Objects, NWC Memo Reg. 4055-34-73, 1973.

7. A. Rosenfeld, "A Non-Linear Edge Detection Technique", Proc. of the IEEE, pp. 814-816, May 1970.

8. R. L. Fox, Optimization Methods for Engineering Design, Addison-Wesley, Reading, Mass., 1971.

9. E. O. Brigham, The Fast Fourier Transform, Prentice-Hall, Englewood Cliffs, N. J., 1974.

10. J. D. Kendrick, and others, "Estimation of Aircraft Target Motion Using Pattern Recognition Orientation Measurements," presented at the IEEE Decision and Control Conference, San Diego, CA., December 1978.

11. M. D. Levine, "Feature Extraction: A Survey", Proceedings of the IEEE, Vol. 57, pp. 1391-1407, August 1969.

12. I. E. Abdou and W. K. Pratt, "Quantative Design and Evaluation of Enhancement/Thresholding Edge Detectors", Proceedings of the IEEE, Vol. 67, pp. 753-763, May 1979.

13. R. L. T. Hampton, A Guided Tour of Kalman Filtering, NWC Technical Note 4070-58-75, 1975.

14. By: International Mathematical and Statistical Libraries, Inc., Houston, Texas, IMSL Library 2, Reference Manual, pp. FTKALM-1 - FTKALM-4, Edition 6, 1977.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor H. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	3
5.	Robert H. Williams Naval Weapons Center Code 62201 China Lake, California 93555	5

187022
Thesis
W618 Williams
c.1 Two dimensional
matched filtering for
attitude measurement
with applications to
target tracking.